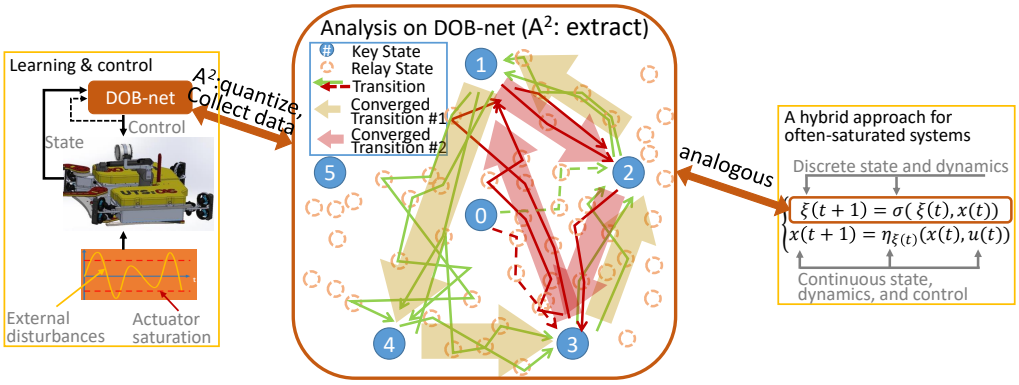
Graphical Abstract

A^2: Extracting Cyclic Switchings from DOB-nets for Rejecting Excessive Disturbances

Wenjie Lu[†‡*] and Dikai Liu[‡]



Highlights

A^2: Extracting Cyclic Switchings from DOB-nets for Rejecting Excessive Disturbances

Wenjie Lu[†‡*] and Dikai Liu[‡]

- Proposed an Attention-based Abstraction (A^2) approach to analyze a Disturbance OBserver network (DOB-net) that actively rejects excessive external disturbances.
- Quantized and abstracted the learned DOB-net via A^2 and then obtained a key Moore machine network that partially reveals the interplay between the learned control strategy and disturbances.
- Found switching mechanisms in the resultant control for rejecting various unobservable (in a statistical sense) disturbances.
- Analyzed the captured switching mechanisms via an analogy to hybrid approaches for often-saturated systems and found that the discrete-event subsystem can be obtained by the proposed A^2.

A^2: Extracting Cyclic Switchings from DOB-nets for Rejecting Excessive Disturbances

Wenjie Lu[†‡*] and Dikai Liu[‡]

Abstract

Reinforcement Learning (RL) is limited in practice by its poor explainability, which is responsible for insufficient trustiness from users, unsatisfied interpretation for human intervention, inadequate analysis for future improvement, etc. This paper seeks to partially characterize the interplay between dynamical environments and a previously-proposed Disturbance OBserver net (DOB-net). The DOB-net is trained via RL and offers optimal control for a set of Partially Observable Markovian Decision Processes (POMDPs). The transition function of each POMDP is largely determined by the environments (excessive external disturbances). This paper proposes an Attention-based Abstraction (A^2) approach to extract a finite-state automaton, referred to as a Key Moore Machine Network (KMMN), to capture the switching mechanisms exhibited by the DOB-net in dealing with multiple such POMDPs. A^2 first quantizes the controlled platform by learning continuous-discrete interfaces. Then it extracts the KMMN by finding the key hidden states and transitions that attract sufficient attention from the DOB-net. Within the resultant KMMN, three patterns of cyclic switchings (between key hidden states) are found, and saturated controls are shown synchronized with unknown disturbances. Interestingly, the found switchings have previously appeared in the control design for often-saturated systems. They are interpreted via an analogy to the discrete-event subsystem of hybrid control.

Keywords: reinforcement learning, finite-state machine, disturbance

[⋆]This work was supported in part by the Australian Research Council Linkage Project (LP150100935), the Roads and Maritime Services of NSW, and the Centre for Autonomous Systems (CAS) at the University of Technology Sydney (UTS). [‡]School of Mechanical Engineering and Automation, Harbin Institute of Technology (Shenzhen), China. [‡]CAS, UTS, Australia. [*]Corresponding author, wenjie.lu@outlook.com

rejection, multiple POMDPs, hybrid system

1. Introduction

Recent advances in deep neural networks have enabled Reinforcement Learning (RL) to solve complex problems. Model-free RL algorithms [40, 42, 25, 49, 24, 34, 39, 50] have shown their success in finding optimal control when it is difficult to precisely characterize all key elements of the targeted problems. However, the usage of RL in practical robotic applications is limited by its grey-box nature. Analytical understanding and interpretation of the learned control networks (policies) remain unsatisfactory, particularly of those acting in continuous state, observation, and control spaces. Compared to the classical analysis of controllers and controlled systems, the learned control networks are missing explainable control mechanisms, analytically-proved stabilities, or guaranteed asymptotical performance. In addition, there is no satisfied representation of the network internal states for fault detection and human intervention, or more importantly for knowledge distillation and transfer [4].

Many efforts have been devoted to explainable neural networks [26, 48]. Recently, finite-state representations of the learned control networks for Atari games have been studied [33], where each game can be viewed as a Partially Observable Markovian Decision Process (POMDP). While a control strategy should be able to adapt to various environments (described by a set of multiple POMDPs) online. This philosophy has found its roots in the existing literature of control designs, such as robust control [51], adaptive control [2, 35, 36], sliding mode control [16], H-infinity control [15]. Understanding the interplay between the environments and the controlled systems is essential for theoretical analysis and a further improvement in control design.

Therefore, this paper focuses on understanding a learned recurrent control network that is able to solve a set of multiple POMDPs, where the transition function of each POMDP is determined by an environment. The recurrent control network has to be aware of environments for effective control. In particular, we are interested in partially understanding the dynamical interplay among environments, environment awareness, and control strategies (captured in the RL-learned control network).

To this end, this paper studies a previously-proposed recurrent policy, i.e., Disturbancec OBserver net (DOB-net) [56] for regulating the position of a

free-floating underwater platform. The platform is further subject to limited actuator (thruster) capacities and various excessive external disturbances. The disturbance and its role in determining the POMDP have to be estimated online for effective and active disturbance rejection. This kind of position regulation problems arise from shallow water applications, e.g., inspecting bridge pile [59], and deepwater operations, e.g., steering a cap to a spewing well [46]. Such problems also exist in controlling quadrotors for surveillance and inspection in windy conditions [57].

Following "less is more" from a poem by Robert Browning, this paper proposes an Attention-based Abstraction (A^2) approach for extracting key memory states and state transitions that reflect the interplay between the DOB-net and the dynamical environments (i.e., disturbances). The proposed A^2 aims to equivalently present the controlled platform as a finite-state automaton. The A^2 extends the Quantized Bottle Network Insertion (QBNI) [33] to the control problems that are better described by multiple POMDPs. The proposed A^2 has two critical improvements to the QBNI approach, as introduced below. Note that in the remainder of this paper, the terminologies "action" and "control" are used interchangeably.

Contributions:

(1): The proposed A^2 first builds a discrete representation of the controlled platform by learning optimal continuous-discrete interfaces for observation and control, respectively. Since the DOB-net operates in continuous spaces, the interfaces between the discrete and the continuous spaces are required for generating KMMN. Instead of manually setting quantization levels, the A^2 learns a more compact quantization that brings about minimum DOB-net performance loss. In this paper, these interfaces are approximated by autoencoders and are optimized with the attention on the subspace where the optimally-controlled platform visits.

(2): A simple recursive loss function is proposed to train an autoencoder for quantizing hidden states, which are key in memorizing and distilling the history of the observations and controls. We found that the autoencoder trained by the recursive loss results in a DOB-net that performs more consistently under various disturbances than the one trained by [33].

(3): The proposed A^2 then creates a Moore Machine Network (MMN) via Partial Enumerative Solution (PES) for minimizing sequential switching functions [45]. After that, A^2 selects the MMN states and the transitions that attract sufficient attention from the DOB-net in solving multiple POMDPs, resulting in a Key MNN (KMMN). The attention that each state attracts

is defined as the number of POMDPs that visit this state. A state only visited by one POMDP is not critical to other POMDPs and is ignored in the KMMN, reducing the complexity of KMMN.

(4): Within the obtained KMMN, we found that about 70% of tested episodes exhibit cyclic transitions between some KMMN states. Also, we found each KMMN state corresponds to a saturated control. This finding is coherent with the fact that often-saturated systems can be described by switching-control-regulated models [65, 6, 62, 14]. It is found that the learned control network activates a portion of the KMMN according to the disturbance pattern, which is formally defined in Section 3.

In this paper, some related work is shown in Section 2. Section 3 introduces the problem of the position regulation, followed by the study scope. Our DOB-net is summarized in Section 4. Section 5 describes the proposed A^2 approach. Then, Sections 6 and 7 present the obtained switching mechanisms and its analysis via an analogy to hybrid control. The last section provides conclusions and future work.

2. RELATED WORK

2.1. Disturbance Rejection

Disturbance rejection control [51, 2, 36, 16] often assumes disturbances bounded and relatively smaller than the control saturation [23]. One popular improvement to these controllers is to add a feedforward compensation based on some disturbance estimation techniques [60]. Various disturbance estimation has been proposed and practiced, such as Disturbance OBserver (DOB) [43, 10, 53], unknown input observer in disturbance and friction accommodation control and nonlinear servo regulation [30, 55, 54], and extended state observer [27, 22]. However, these controllers fail to guarantee stability considering the actuator saturation [21] when disturbances frequently exceed control saturation.

To this end, model predictive control (MPC) [9] is often applied due to its capability in dealing with constraints [21]. It formulates a series of constrained optimization problems over receding time horizons based on predictions of the disturbed platform. A prediction method (e.g., autoregressive moving average) is required to forecast future disturbances based on the estimations of current disturbances (from DOBs). However, DOBs often require sufficient system modelling, which could be difficult for underwater robots due to hydrodynamic effects. Current DOBs might not have the insufficient

capability in estimating fast time-varying disturbances; their convergence analysis often assumes disturbances time-invariant. In addition, such separated processes of disturbance estimation, disturbance prediction, and control optimization might not be able to produce estimations and control signals that are mutual robust to each other, as evidenced in [7, 31].

2.2. Q-Learning

RL (also known as iterative learning, adaptive dynamic programming, and neural computing) has drawn a lot of attention in finding optimal controllers for systems that are difficult to model accurately. Q-learning is a widely-used model-free reinforcement learning approach, its goal to train a policy that outputs action given system state. It does not require a model of the environment or system, it can adapt to stochastic transitions and rewarding mechanism. Recently, deep RL algorithms based on Q-learning [40, 1], policy gradients [49, 24], and actor-critic methods [34, 39] have been shown to learn very complex skills in high-dimensional state and action spaces, including simulated robotic locomotion, driving, video game playing, and navigation.

2.3. Hybrid System

A hybrid system describes a set of collaborative agents or subsystems. It is often represented by multiple modes of dynamics, which are chosen by discrete actions or events [52]. Therefore, a hybrid system is characterized by discrete and continuous control and state, together with continuous-discrete state interfaces. It has been used to control centralized multiple agents in [17], and decentralized networks in pursuit-evasion games [63]. A comprehensive introduction can be found in [8].

2.4. Understanding Recurrent Policy Networks

Recurrent Neural Network (RNN) memory (i.e, hidden states) is often in the form of a high-dimensional vector in a continuous space and is recursively updated through gating networks. There has been some work on visualizing and understanding the learned RNN [32]. RNN models have been linked to iterated function systems in [3], which further shows the relationship between the independent constraints on the state dynamics and the universal clustering behavior of the network states. Many others use training data to show the clustering and the correspondences between network internal states

[12]. An additional deep neural network has been trained to better visualize the correlation between inputs and outputs [61].

There has been a number of research on extracting finite-state machines from trained RNNs. Crutchfield has reported that the minimal finite-state machine could be induced from periodic sampling with a single decision boundary [13]. An approach that forces the learning process to develop automaton representations has been proposed in [20], which adds a regularization to constrain the weight space. Omlin has used hints to learn a finite-state automaton for second-order recurrent networks [44]. Learning full binary networks is an orthogonal effort [29] to the previously mentioned, where activation functions (and/or weights) are binary. A query-based approach has been proposed to extract a deterministic finite-state machine that characterizes the internal dynamics of hidden states [58].

Koul has proposed Quantized Bottleneck Network (QBN) insertion in [33] for extracting a finite-state machine from discrete action networks. The QBNs are autoencoders, where the latent encoding is quantized. Given a trained RNN policy, the QBNs are trained to encode and quantize the hidden states and observations in a supervised manner.

3. CONTROL PROBLEM and SCOPE

The optimal control problems considered in this study involve a free-floating platform (a rigid body) under translational motion, thanks to its huge restoring forces and its sufficiently large torque capacity on the heading control. The position of this platform is denoted as $q \in \mathbb{R}^3$. The platform's velocities and accelerations are denoted by $\dot{q}$ and $\ddot{q} \in \mathbb{R}^3$, respectively. It is assumed that q and $\dot{q}$ are observable without errors in this study, nevertheless, RL approaches are in general robust to reasonable observation noises.

Then the platform's dynamics (also referred to as the system) is given by

$$M(q)\ddot{q} + g(q) = u + d, \tag{1}$$

where $M(q) \in \mathbb{R}^{3\times3}$ is the inertia matrix and $g(q) \in \mathbb{R}^3$ is the vector of the gravity and buoyancy forces. This matrix, vector, and external disturbances $d \subset \mathbb{R}^3$ are assumed unknown to the controller (the trained DOB-net). The platform control $u \in \mathcal{A} \subset \mathbb{R}^3$ is saturated at an upper bound $u^- = \max(\mathcal{A}) \in \mathbb{R}^3$ and a lower bound $u_- = \min(\mathcal{A}) \in \mathbb{R}^3$, where max and min are dimension-wise operators. Let $x = [q^T, \dot{q}^T]^T \in \mathcal{X} \subset \mathbb{R}^6$ and the platform dynamics in

discrete time can be written as

$$x(t+1) = f\big(x(t), u(t), d(t)\big). \tag{2}$$

In the remainder of this paper, the time indices t in equations are in parentheses and the ones in figures are subscripts for compactness.

Disturbance and pattern: The external disturbances d are described by the disturbance forces, which are time-variant and are superpositions of l sinusoidal functions as

$$d(t) = \sum_{1 \le i \le l} d_i(t), \tag{3}$$

where $d_i(t) = A_i \sin(w_i t + \phi_i)$, and l denotes the number of components, which is unknown and may vary across environments. The parameters (A_i, w_i, and ϕ_i) of each component d_i are assumed uniformly and randomly sampled from given intervals and then fixed in each environment in this paper. One instantiation of all parameters of all l components is referred to as one *disturbance pattern*.

In the remainder of this paper, one sampled disturbance pattern is viewed as one environment to the free-floating body. The terms "disturbances" and "disturbance forces" are used exchangeably. Note that the external disturbances considered are excessive to the free-floating platform, the definition of which is given as follows.

Definition 1 (Excessive external disturbances). *Excessive external disturbances are those defined in Eq. (3), where the amplitudes (A_i) exceed the platform control saturation (u^- and u_-).*

Problem 1 (Optimal control). *Find one controller that chooses an action $u(t)$ for the system described in Eq. (2) at time t in response to the current observation $x(t)$, such that the discounted summation of collected rewards is maximized. The summation is expected over episodes and is defined as $\mathbb{E}\sum_{\tau=t}^{T-1} \gamma^{\tau-t} r\big(x(\tau), u(\tau)\big)$, where $r(\cdot)$ is a reward function (additive inverse of the tracking error, defined later), T denotes the number of steps in an episode, and $\gamma \in [0, 1)$ is a discount factor that prioritizes near-term rewards over future rewards [41].*

The tracking error is defined as

$$\eta(t) = \|x(t)\|, \tag{4}$$

where $\|\cdot\|$ denotes the L2 norm. In each episode, the environment is randomly sampled and is characterized by excessive disturbances in Eq. (3).

Classical RL approaches often implicitly assume $d(t)$ independently identically distributed (i.i.d.), possibly conditioned on the platform state x [47]. If not conditioned on x, $d(t)$ is marginalized over t and x, and is then described as $d(t) \sim \mathcal{N}(\epsilon, E)$, where $E = \text{diag}\,(\sigma_1^2, \ldots, \sigma_D^2)$. These disturbance models lead to a single-POMDP description of the controlled platform and are sufficient when disturbances are small. However, the excessiveness makes these models of d not suitable for disturbance rejection, as evidenced in [56]. The following analysis shows that the controlled systems in Problem 1 are better described by multiple POMDPs.

For a jth pattern of disturbance superposition, each component $d_{i,j}(t)$ of $d_j(t)$ is a function that exhibits periodicity, which can be described as a Markovian chain. The index j is dropped if no ambiguity is caused. The Markovian chain is given as

$$\begin{bmatrix} d_i(t+1) \\ \dot{d}_i(t+1) \end{bmatrix} = g_{i,j}\Big(\begin{bmatrix} d_i(t) \\ \dot{d}_i(t) \end{bmatrix} \Big),\ 1 \le i \le l, \tag{5}$$

where the index $j \in \{1, 2, \cdots, \infty\}$ of $g_{i,j}$ indicates the variety of disturbance patterns. Let $\mathcal{D} \subset \mathbb{R}^6$ denote the space of $[d_i^T(t+1), \dot{d}_i^T(t+1)]^T$ and $\mathcal{G}$ the space of possible $g_{i,j}$.

Let $z_j = [x^T,\ d_{1,j}^T,\ \dot{d}_{1,j}^T,\ \cdots, d_{l,j}^T,\ \dot{d}_{l,j}^T]^T \in \mathcal{Z}_j = \mathcal{X} \times \mathcal{D}^l$, where l might vary across environments. Then the platform dynamics can be rewritten in a partially observable Markovian chain as

$$\begin{aligned} z_j(t+1) &= f_j\big(z_j(t), u(t)\big), \\ y(t) &= h_j\big(z(t)\big) = x(t), \end{aligned} \tag{6}$$

where $h_j(\cdot)$ is the observation function, showing that $x(t)$ is observable while $d(t)$ is not directly observable. Here the observability is in a statistical sense (not in a control sense). Let $\mathcal{F}$ denote the space of all possible f_j. Each transition function f_j defines a POMDP $P_j = \{\mathcal{Z}_j, \mathcal{A}, f_j, h_j, \pi\}$, where π is the trained current control network. Let $\mathcal{P}$ denote the set of all possible P_j.

The control network π is targeted to solve Problem 1 (i.e., all $P_j \in \mathcal{P}$). Key to π is the integration of a disturbance observer to existing RL frameworks. This observer not just estimates the unobservable state $d_j(t)$ but also infer the transition function f_j. Both $d_j(t)$ and f_j are critical to the control subnetwork. Our previous work has proposed a DOB-net for this purpose [56]. The DOB-net outperforms existing control and RL approaches.

However, the understanding of the learned DOB-net remains unsatisfactory. Therefore, the scope of this paper, shown below, is regarding the understanding of the learned DOB-net. For simplicity, the reduced version of the DOB-net is studied in this paper.

Scope 1 (Analysis of DOB-nets). *Inductive reasoning of the mechanism on how the learned DOB-net responds to different unobservable external excessive disturbances (i.e., to different POMDPs).*

4. DOB-NET

Estimating the disturbance forces, their transition functions, and their predictions is key in solving a P_j randomly sampled from $\mathcal{P}$. In DOB-nets, these estimations are encoded in a latent feature space. The features have to be mutual robust between the controller and the observer. The DOB-net developed in our previous work is composed of a disturbance-behavior observer subnetwork and a controller subnetwork. For simplicity, this paper investigates the reduced version consisting of a single-layer GRU, as shown in Fig. 1. Both subnetworks are jointly optimized for mutual robustness and unified optimization. The observer subnetwork imitates the classical DOB mechanisms and is enhanced with the flexibility from GRUs, instead of only providing the estimation of the lumped disturbances up to the current time. The encoding h_t (shown in Fig. 1) is supposed to represent the disturbance behavior that is key to controller subnetwork.

The full DOB-net is constructed based on the classical actor-critic architecture [38], the network outputs actions and critics (also referred to as cost-to-go) associated with previous state and action. The policy is trained using simulated sine-wave disturbances. Multiple control and RL algorithms have been tested and compared in [56], the results have demonstrated that the proposed DOB-net does have a significant improvement in rejecting excessive disturbances. In fact, this DOB-net closely relates to meta reinforcement learning (meta-RL) [18], which often considers a distribution of tasks and the tasks differ in transition models or reward functions. Meta-RL is interested in a framework that leverages data from previous tasks to acquire a learning procedure that can quickly adapt to new tasks.

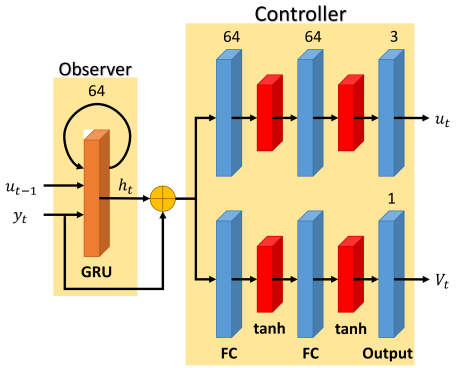


Figure 1: Network architecture of DOB-Nets. FC: fully connected layer.

5. A^2: Extracting Key Moore Machine Network

The proposed A^2 approach aims to abstract the control mechanism captured in the trained DOB-net for solving continuous-control problems. It consists of two procedures: quantization and abstraction. The A^2 has two critical improvements to the "Quantized Bottleneck Network Insertion" (QBNI) [33], the latter of which is used to generate a finite-state automaton of a trained policy network.

Definition 2 (Finite-state automaton [11]). *A finite-state automaton is an abstract machine whose state is assigned as one of a finite number of states at any given time. It is also referred to as a finite-state machine. The transition between states is determined by discrete action and observation, which is often given by a table.*

The finite-state automata in this paper are all deterministic.

Definition 3 (Moore machine network [33]). *A Moore machine network is a standard deterministic finite-state machine whose states are labeled by their output values (controls in this paper). An MMN is fully characterized by finite sets of states, observations, and actions, a transition function, and a policy that maps states to actions, where the policy and the transition function are represented by neural networks.*

The QBNI algorithm together with the PES works well for grouping hidden states and observations (and thus reducing the number of states in an

MMN). However, the effectiveness of the PES heavily depends on the number of actions, which has to be limited. At least one state is related to a unique action [45], therefore the number of possible actions has to be reduced for revealing the interplay in Scope 1. In the cases of Atari games, the possible actions are often fewer than 8 (e.g., "fire", "move left/right", "jump"). As pointed in the introduction, the problems studied here involve multiple POMDPs, leading to a large number of states and transition in the obtained MMN.

The first improvement is in the quantization, where continuous-discrete interfaces are optimized for actions, reducing the number of quantized actions given acceptable DOB-net performance loss. The second improvement is the abstraction of key states and transitions in the MMN based on the evaluation of attention.

5.1. Continuous-Discrete Interfaces

The proposed A^2 approach first learns continuous-discrete interfaces for observations and action, respectively. From the perspective of hybrid control [37], the continuous-discrete interfaces offers essential connections between continuous states and discrete modes. The switchings between discrete modes build an automaton that provides an interpretation of the switching mechanisms found later. From the perspective of machine learning, these continuous-discrete interfaces become an autoencoder with a quantization layer as the encoding layer.

The observation and action interfaces in the quantized DOB-net have been shown in Fig. 4 (better viewed in color), which are denoted as Observation Quantization (OQ) and the Action Quantization (AQ), respectively. Each Quantization block consists of a continuous-to-discrete interface and a discrete-to-continuous interface.

Then, the components in the blue dashed rectangle and the ones in the green dash-dotted rectangle correspond to the discrete-event subsystem and the mapping from the discrete hidden state to the continuous control, respectively. This will be discussed more in Section 6. In this paper, all interfaces are built upon neural networks, the detailed structures of which are shown in Section 6. In fact, each quantization block is an autoencoder from the perspective of machine learning.

In general, autoencoders consist of an encoder and a decoder, where the decoder aims to reconstruct the original inputs to the encoder. The au-

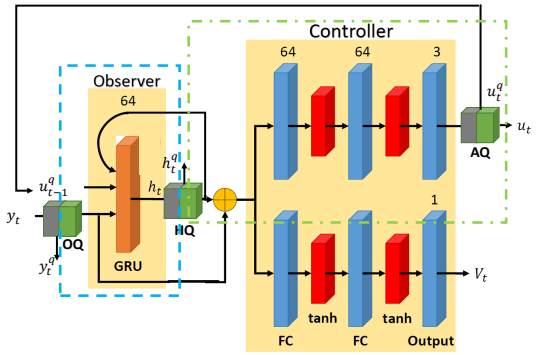


Figure 2: Network architecture of the quantized DOB-Net.

toencoder has been used widely to reduce the data dimension using neural networks [28], which is often trained in a supervised manner.

One straightforward approach to have the interfaces is to evenly quantize the observation and action space, however, the quantization levels are not clear. Also, the importance of action (observation) to the controlled platform is not uniform across the action (observation) space. The states and actions that attract the most attention from the optimally-controlled platform are often subsets of the entire state and action spaces, respectively. We are interested in the interfaces that are both optimized with respect to these subsets.

In this paper, to produce a continuous-to-discrete interface, the output of the encoder is quantized through a combination of a 3-level activation layer (denoted as Tanh*) and a quantization layer. Same to [33], the Tanh* layer restricts the outputs in the range of $[-1, 1]$ and offers 0 gradients near a 0-valued input, which allows a quantization level at 0 during training. The Tanh* activation function is given as [33]

$$\phi(x) = 1.5\tanh(x) + 0.5\tanh(-3x). \tag{7}$$

With Tanh*, the quantization layer offers 3-level quantization valued at $\{+1, 0, 1\}$.

With the continuous-discrete interfaces inserted, the full quantized DOB-net is illustrated in Fig. 2. In the remainder of this paper, the original DOB-net is referred to as "continuous DOB-net" to distinguish from the quantized DOB-net.

Training: The QBNI algorithm, suggested in [33], does not work well

for learning OQ and AQ, since the number of quantized actions should also be minimized for effective reduction in obtaining a key MMN. Therefore, a three-step training approach is used to train both OQ and AQ. The number e of neurons in the encoder layer of AQ determines the cardinality of the set of all possible discrete actions. The cardinality is 3^e since each quantization neuron has 3 levels. On one hand, a large e leads to less optimality loss from the quantization, compared with the continuous DOB-net. On another hand, a small e results in fewer action choices and thus fewer MMN states after reduction by PES. Therefore, the number of discrete actions is expected to be minimized for the benefit of reducing the number of states in the MMN. By choosing the number of neurons in the quantization layer, the performance degeneration should be restricted within a reasonable number (e.g., 10%).

Step one: The continuous DOB-net is first trained by the Advantage Actor Critic (A2C) [39], as shown in [56]. A2C uses synchronous gradient descents for optimizing policy networks and it executes multiple instances of the environments in parallel threads. This parallelism provides a more training estimation of critics.

Step Two: A data set of observations and actions from a large number of episodes is collected through using the trained continuous DOB-net. Note that in each episode, a disturbance pattern is randomly generated, which is i.i.d. to the pattern in another episode. Then OQ and AQ are trained respectively using the observation and action data through supervised learning. Since the data is collected from using the optimal DOB-net, the data reflects the nonuniform distribution of attention in the action and observation space.

Step Three: The trained OQ and AQ are inserted into the trained continuous DOB-net to obtain the quantized DOB-net, as shown in Fig. 2 (HQ is deactivated). However, the performance of the quantized DOB-net is not close to the continuous DOB-net (worse by 31%). Then the entire quantized DOB-net is finetuned in an RL fashion, same to Step One. The quantization layer introduces functions that are non-differentiable. During the training, a straight-through estimator for gradients, as suggested in [5], is adopted. The estimator simply treats the quantize function as an identity function during backpropagation and passes on the gradients without any change. The results shown in Section 6 suggest that the performance of the quantized DOB-net resulted from the three-step training is close to the performance from the continuous DOB-net.

Once the quantized DOB-net (HQ deactivated) is obtained, a data set of hidden states is collected and used to train HQ, as in [33]. With the trained

HQ insertion as illustrated in Fig. 2, the full quantized DOB-net is available.

5.2. Key Moore Machine Network

The data sets of the discrete hidden states, the discrete observations, and the discrete actions are collected during solving Problem 1 in multiple randomly-generated environments. In addition, the transitions between consecutive pairs of the quantized hidden states are also recorded.

Then unique states are found and indexed for each data set, resulting in an MMN. Let m denote the cardinality of the state space of the MMN and n the cardinality of the observation space of the MMN, then the transition function of this MMN is constructed as a transition matrix of $n \times m$ that captures the transitions evidenced in the data. In general, m and n are larger than necessary.

A reduced but equivalent MMN can be obtained by a standard finite state machine reduction technique (i.e., PES in this paper), which is able to group hidden states and observations if a common transition and action can be found. Each group of the hidden states is referred to as a state in the reduced MMN and each group of the observations is referred to as an observation in the reduced MMN. This reduced MMN is able to show how states, observations, and actions are related to problems, as shown in [33].

However, Problem 1 subject to various environments are better described by multiple randomly sampled POMDPs. The number of states and observations in the reduced MMN are still too large to induce explainable relationship among states, action, and environments. In fact, the systems (controlled by the quantized DOB-net) visit different portions of the reduced MMN in different episodes (i.e., under various disturbance patterns), as illustrated in Fig. 3. As shown in Section 6, the number of states in the reduced MMN was still quite large (91) compared to Atari games investigated in [33].

In order to understand the interplay between disturbances and control strategies, in this paper, we propose a Key Moore Machine Network (KMMN), which ignores some states and transitions in the reduced MMN. Some of the states and observations are unique to an episode (i.e, a POMDP), while others attract more attention from a number of episodes.

Definition 4 (Key Moore machine network). *A key Moore machine network is a finite-state automaton that only consists of the key states and transitions between key states. The key states are those MMN states that attract sufficient attention from the controlled systems in a number of environments.*

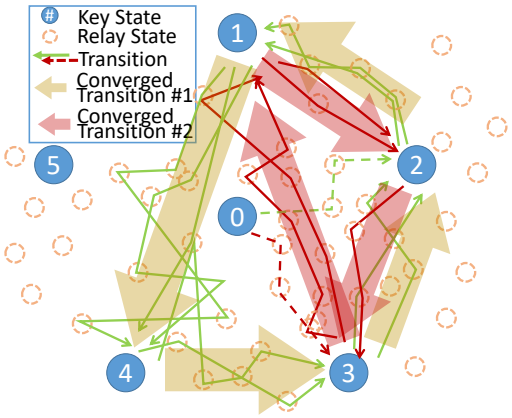


Figure 3: Key transitions; only a portion of the reduced MMN is shown.

The attention of a state is defined as the number of episodes that visit this state. A transition between the key states is available if a concatenated transition can be found in the reduced MMN.

The relation between the KMMN and the reduced MMN is shown in Fig. 3, where the MMN states other than key states are referred to as relay states. Given two POMDPs, the set of the visited key states and the relay states is only a subset of all nodes in the reduced MMN. Therefore only a portion of the reduced MMN is shown. Note that the transitions in KMMN are different from the MMN transitions. A KMMN transition may involve multiple MMN transitions. One MMN transition corresponds to one step defined in POMDPs. Since we are interested in the interplay between the control strategies and the environments (i.e., disturbances, POMDPs), we extract key MMN states that are commonly visited by a number of POMDPs. These POMDPs share some similar properties, for example, periodicity and excessiveness. The common states link to these similar properties and offer some insights on the interplay mechanism. The KMMN greatly reduces the number of states and transitions, providing a baseline for inductive learning of the interplay. To find KMMN, the step of obtaining the reduced MMN is necessary. Otherwise, the chance of having states with sufficient attention is quite low.

6. IMPLEMENTATION and RESULTS

This section first outlines the simulation details of the platform and disturbances. Then, the implementation of learning interfaces (AQ, OQ, and HQ) and results are presented. After that the obtained MMN and KMMN are summarized, as well as the found switching mechanism captured in the DOB-net.

6.1. Platform and Disturbances

As described in the problem formulation, the platforms are assumed stable in orientation. Only translational motion and control are considered, thus, the platform has a 6-dimensional state space (positions and linear velocities) and a 3-dimensional action space. In order to analyze the results more intuitively, the characteristics (mass, control, gravity and buoyancy forces, and disturbance forces) of the platform are scaled down such that the mass of the simulated platform is 1 [kg]. Then, the control saturation is given as $u^- = -u_- = [2,\ 2,\ 2]^T\ [N]$.

Each episode contains 200 steps with 0.05 second per step. In each episode, the platform starts at a random position with a random velocity, and it is controlled to reach a given position (the origin), aiming to keep its position within a range (as small as possible) to the origin against unknown excessive disturbances. In these simulations, the external disturbances are exerted on all three directions in the inertial frame. In each axis, the disturbance is sinusoidal and then the disturbance superposition is given as

$$d(t) = \begin{bmatrix} A_x \sin(\frac{\pi}{T_x}t + \phi_x) \\ A_y \sin(\frac{\pi}{T_y}t + \phi_y) \\ A_z \sin(\frac{\pi}{T_z}t + \phi_z) \end{bmatrix}, \tag{8}$$

where

$$\begin{aligned} A_x, A_y, A_z &\sim U(2.6, 3) \\ T_x, T_y, T_z &\sim U(2, 4) \\ \phi_x, \phi_y, \phi_z &\sim U(-\pi, \pi), \end{aligned} \tag{9}$$

and $U(a, b)$ denotes a uniform distribution in the range $[a,\ b]$. According to the problem setting, the amplitudes of disturbances exceed the control limits by $30\% - 50\%$. The purpose of the DOB-net training is to enable the trained network to deal with unknown time-varying disturbances, thus the values of

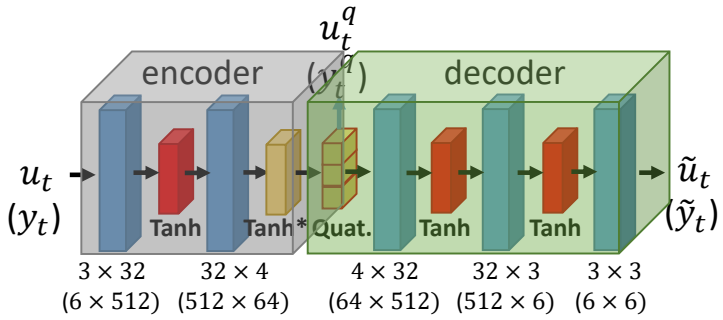


Figure 4: Network structure of AQ and OQ. Quantities in parentheses correspond to OQ.

the amplitude, period, and phase are randomly sampled in each training or testing episode. The component in each axis is sinusoid; this setting offers a better illustration to understand what control mechanism the network policy has learned.

6.2. Learning Interfaces

The interfaces for action (AQ) are illustrated in Fig. 4, which consists of 5 linear layers, 1 quantization layer, and 4 hyperbolic tangent (denoted as Tanh) activation layers. One of the activation layers is a 3-level activation layer (defined in Eq. (7) and denoted as Tanh*). The encoder component of the autoencoder is a continuous-to-discrete interface, while the decoder component is a discrete-to-continuous interface. The interfaces for action and observation share a similar autoencoder structure with different numbers of neurons in linear layers and the quantization layer. In Fig. 4, the numbers and symbols in parentheses show the input, the output, and the number of neurons regarding OQ.

The neuron numbers were manually picked such that the quantized DOB-net performs similarly to its continuous counterpart. As pointed out earlier, the number of neurons in the encoding layer of AQ is critical. It is expected to minimize this number without losing much optimality in the resultant quantized DOB-net. It was manually picked via the trial-and-error approach. The neuron number was first set to 3, however, the resultant performance was not satisfactory. The collected reward (negative) was nearly doubled. Then, the neuron number was set to 4 and 5, respectively. It was found that 4 is sufficient for retaining optimality. The continuous DOB-net and quantized DOB-net exhibit on average 10% difference in rewards collected

in an episode. The number of neurons in OQ is also critical, choices of 32, 48, 64, and 128 were tested and it was found that 64 is appropriate for the DOB-net. The choice of neuron numbers has been studied in the field of neural architecture search and can possibly be solved via RL [64], however, it is out of the paper scope.

Since the disturbances exceed the control saturation frequently, the platform inevitably oscillates and so does the error of position regulation. The DOB-net requires some steps to collect sufficient data to infer the environment in the hidden state. Here the maximum tracking error from Step $t = 150$ to Step $t = 200$ is used as a criterion to show the effectiveness of the learned DOB-nets. It is referred to as the regulation error and given as

$$R = \max_t \eta(t), 150 \leq t \leq 200. \tag{10}$$

The 3D trajectories from both quantized and continuous DOB-nets for the same problems (i.e., same POMDPs defined in Eq. (6)) have been illustrated in Figures 5 and 6, respectively. The transparent red and blue spheres respectively represent the regulation errors from the quantized and continuous DOB-nets. Clearly, the quantized DOB-net was able to achieve trajectories similar to the one from the continuous DOB-net. Furthermore, the regulation error did not increase much.

In addition, Robust Integral of the Sign Error (RISE) control [19] and classical RL [39] were also tested by the same problems. Both approaches resulted in worse performance than the DOB-net, as illustrated in Figures 7 and 8.

6.3. Moore Machine Networks

Once the interfaces for action and observations were trained, another set of simulations using the quantized DOB-net was conducted. A data set of the GRU hidden states was collected from 1000 episodes. In each episode, the disturbance pattern was randomly generated according to Eq. (9). Following [33], the autoencoder for quantizing hidden states is illustrated in Fig. 9, which consists of 6 linear layers, 1 quantization layer, and 6 Tanh activation layers, where one of the activation layers is Tanh*.

The data collected was used to train HQ in a supervised manner. Different from the usual loss functions [33], the importance of recursive stability was emphasized. The loss function used has two terms; the first one is standard

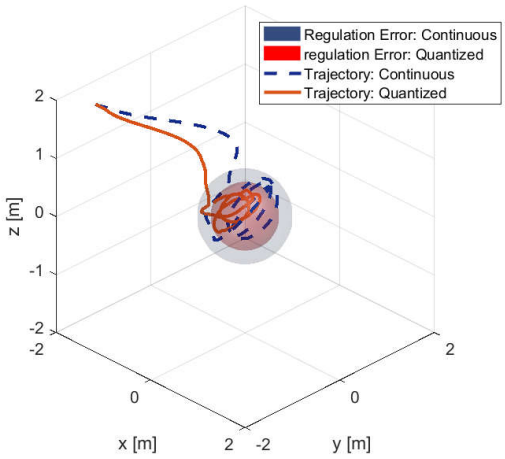


Figure 5: Comparison of example trajectories from the continuous DOB-net and the quantized DOB-net.

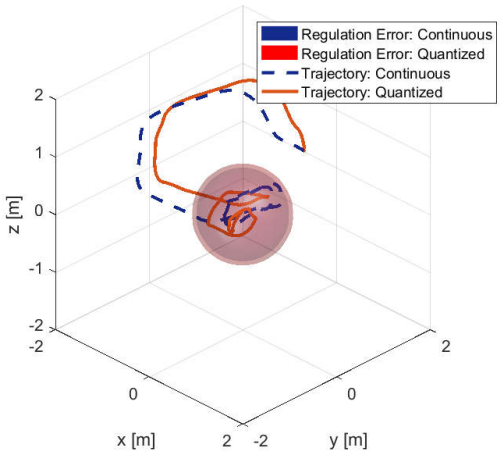


Figure 6: Comparison of example trajectories from the continuous DOB-net and the quantized DOB-net.

and the second one regulates the recursive stability. The loss function L is defined as

$$L = ||h_t - HQ(h_t)|| + \eta||h_t - HQ(HQ(h_t))||, \tag{11}$$

where η was set as 10. Using a stochastic gradient descent approach with the learning rate $1e{-}4$, the training error (mean square error) was $1.2e{-}3$. The HQ network was inserted into the quantized DOB-net, as suggested in

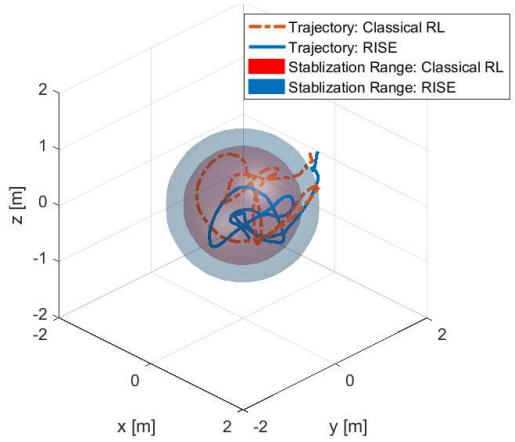


Figure 7: Example trajectories from RISE and the classical RL.

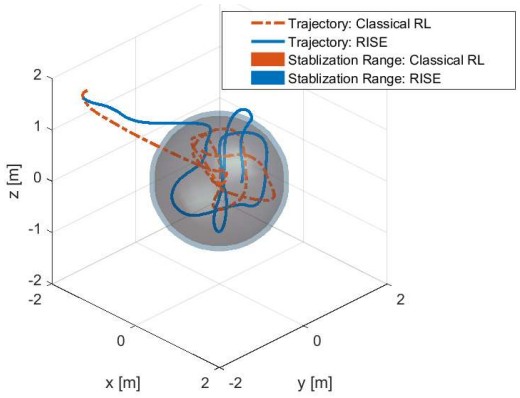


Figure 8: Example trajectories from RISE and the classical RL.

[33], resulting in the full quantized DOB-net. The rewards collected in each episode by the quantized DOB-net has been compared with the ones collected by the continuous DOB-net in Fig. 10, showing about 12% degeneration averaged over all episodes. As shown in Fig. 11, the averaged regulation error exhibited 8% increase.

Then another data was collected from simulations of 400 episodes using the quantized DOB-net (with HQ inserted). Each episode has 200 samples of observations, hidden states, current actions, and previous actions. Also, the transitions between hidden states given observations and actions were recorded. It was found that the number of the unique hidden states was

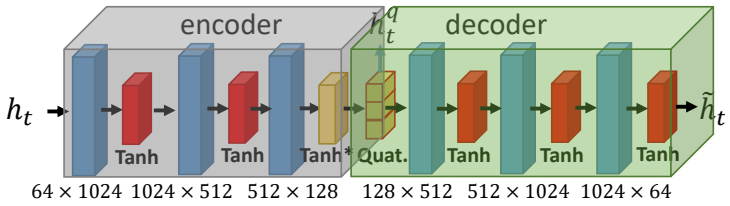


Figure 9: Hidden state quantization.

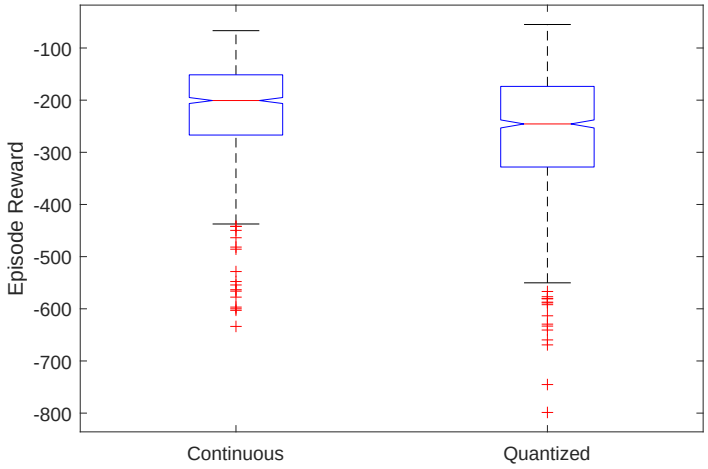


Figure 10: Rewards collected from 1000 episodes.

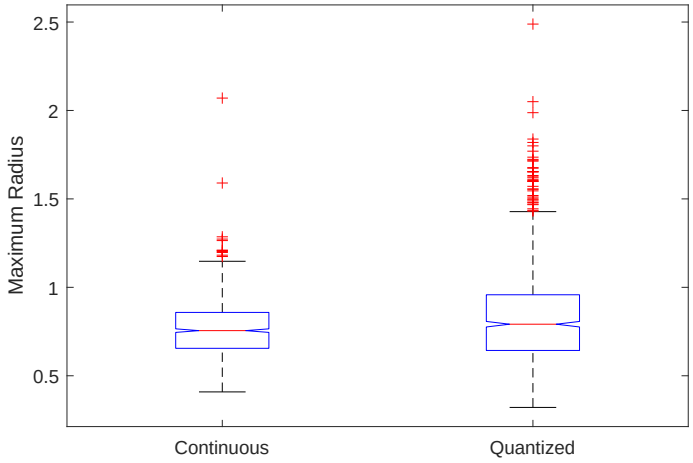


Figure 11: Position regulation errors collected from 1000 episodes.

16931 and the number of the unique observations was 15619, suggesting that the system controlled by the quantized DOB-net in multiple environments did visit a large number of discrete hidden states. The number of the unique actions was 80, the maximum of which is 81.

Considering the transitions between discrete hidden states as an incompletely specified sequential switching function, the number of hidden states and observations was grouped by PES [45]. The number of unique groups of hidden states in the reduced MMN was reduced to 114 and the number of observation groups was 2047. We refer to each of these groups as a state or an observation in MMN. It is nearly impossible to find insights about the interplay between environments and control strategies, due to a large number of transitions and states. A portion of the MMN that highlights the transitions and states visited by two episodes has been illustrated in Fig. 3, where the key states were obtained in the following subsection.

6.4. Key Moore Machine Network

The goal of the KMMN is to extract some shared control logic used by the learned DOB-net to solve different POMDPs defined in Eq. (6), and thus to show the interplay between the control and disturbances. Here data from 20 episodes were studied. The sufficient attention was defined as "85% attention". In other words, being qualified as a key state in KMMN, the state must attract attention from at least 17 episodes out of 20. The episode number was picked to balance the computational complexity and chance of finding the KMMN.

We found that 6 key states were picked by those 20 episodes. One of the key states is the initial state since in all episodes the hidden state always started at zero. The key states found are shown in Table 1, which summarizes the key state indices, the quantized encodings, and the decoded actions. It was found the action at the beginning of each episode was almost zero, while the actions associated with other key states were always at the control saturation. More about this phenomenon will be discussed later in this section.

The transitions between the key states in 14 episodes (out of 20) converged to some cyclic patterns shown in Fig. 12. Figure 12 shows 8 examples, where the first 2 examples did not exhibit clear converged patterns. The remaining 6 examples exhibited three cyclic transition patterns, highlighted by green solid arrows. In all examples, the state started from State 0 and the system took a number of transitions to enter one of the cyclic patterns. It is because at the

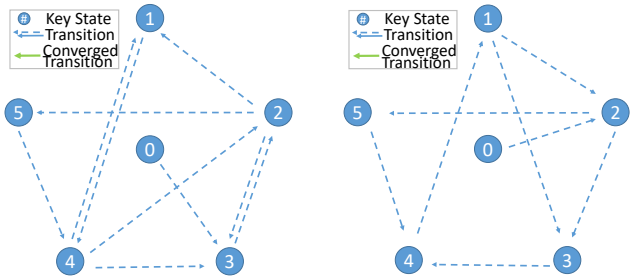
Table 1: Key state description

Key state index	Quantized encoding	Decoded action
0	$[0, \ 0, \ 0, \ 0]^T$	$[0.03, -0.15, \ 0.07]^T$
1	$[1, \ 1, -1, \ 1]^T$	$[2, \ 2, \ 2]^T$
2	$[1, -1, \ 1, -1]^T$	$[-2, \ 2, -2]^T$
3	$[-1, -1, \ 1, -1]^T$	$[-2, -2, -2]^T$
4	$[-1, \ 1, -1, \ 1]^T$	$[2, \ -2, \ 2]^T$
5	$[-1, -1, -1, -1]^T$	$[-2, -2, \ 2]^T$

beginning of each simulation (episode), the DOB-net intended to interact with the environments to gain observations for estimating the key aspects of the inherent POMDPs (i.e., disturbances and their transfer functions). The following analysis partially reveals how the hidden states are related to controls and disturbances.

Considering the associated action with each state in the KMMN, it was found that the learned DOB-net behaved similarly to a hybrid controller where switchings occur. These switchings exhibited cyclic patterns due to the fact the disturbance in each direction was periodic. Each switching pattern indicated a disturbance pattern. As shown in Fig. 13, the disturbances in three directions are illustrated in red, green, and blue, respectively. The additive inversion of the controls associated with the states is also illustrated. Note that the values of the controls in x and z directions were added by -0.2 and 0.2, respectively, for a clear illustration. It was found that the states in the KMMN were only activated when the disturbances were close to the control saturation, as shown in Fig. 13. By inspecting the controls and unknown disturbances, it was shown that the obtained actions were synchronized with the disturbance forces.

Some episodes exhibited similar converged transition patterns, as shown in Fig. 12 (c), (d), (f), and (h). However, the way the system entered into the cyclic patterns varied in different environments. The system is shown in Fig. 12(c) entered the cyclic pattern (referred to as cycle) through State 2 directly, while the system shown in Fig. 12(h) entered the cycle through State 4 after visiting State 3. In Fig. 12(d), the system visited States 3 and 5, and then entered into the cycle at State 4. As illustrated in Fig. 13, in cases of (c), (d), (f), and (h), the disturbance forces in x and y directions had similar frequencies and phases, while the disturbance force in z direction



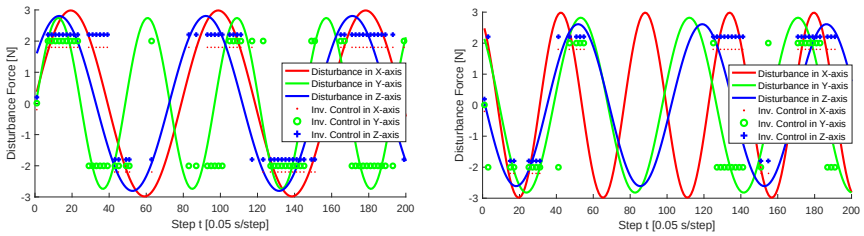
(a) Episode 1 (b) Episode 2

(c) Episode 3 (d) Episode 4

(e) Episode 5 (f) Episode 6

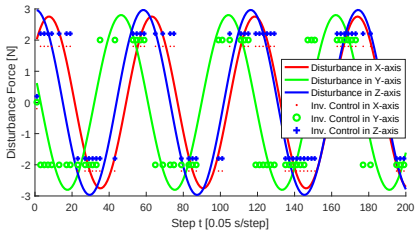
(g) Episode 7 (h) Episode 8

Figure 12: Transitions between key states and cyclic patterns.

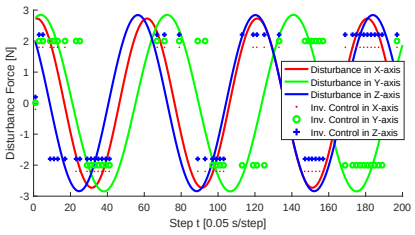


(a) Episode 1

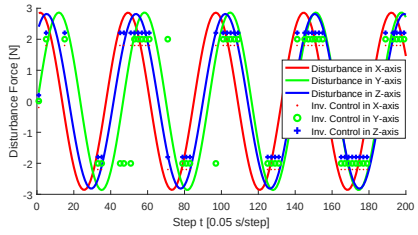
(b) Episode 2



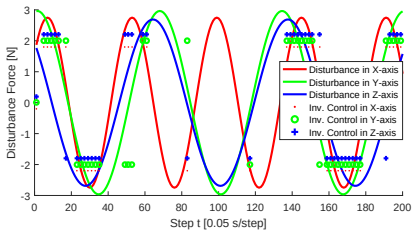
(c) Episode 3



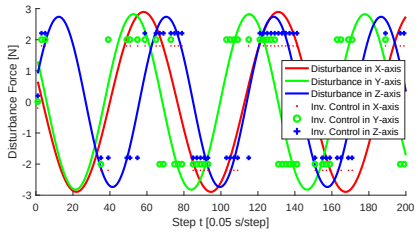
(d) Episode 4



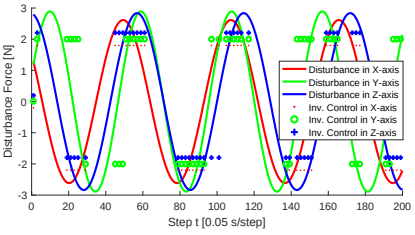
(e) Episode 5



(f) Episode 6



(g) Episode 7



(h) Episode 8

Figure 13: Disturbances and additive inverse controls at key states.

had different a frequency and phase. These examples show that the DOB-net was able to estimate disturbances and their inherent governing behavior.

In Fig. 12(e) and (f), the systems exhibited another two cyclic patterns. Interestingly, the episodes shown in Fig. 12(g) and (e) exhibited two cyclic patterns, respectively. For example, the system in Fig. 12(g) first entered the cycle (shown in blue solid lines) through State 3 and then entered the second cycle at State 4 and stayed in the second cycle.

The first two examples in Fig. 12 did not exhibit clear cyclic patterns. It is possible that the key states found in those 20 episodes did not capture the states that were crucial to these two examples. More research about the definition of sufficient attention should be explored in future research.

The system in Fig. 12(e) entered in a binary switching pattern. With careful examination of the disturbances in Fig. 13(e), we found the components in the randomly-generated disturbances had similar periods and phases. Therefore, the two states in the KMMN were sufficient to capture the periodic shifts. Overall, the key states found have a strong correlation between disturbance patterns and the time instants when the disturbance forces were close to control saturation. The phases between disturbances change as a function of time, as shown in Fig. 13, which strongly ties the change of the hidden states and the action associated. Therefore, the observer designed in the DOB-net and learned together with the control subnetwork was able to estimate such a shift in the phases and magnitudes of the disturbances.

7. DISCUSSION

As pointed in [65, 6, 62, 14], the controlled platform whose control often reaches control saturation can be described by a switching-control-regulated system. This kind of systems can be characterized by

$$\begin{aligned}\xi(t+1) &= \sigma\big(\xi(t), y(t)\big)\\ z(t+1) &= \eta_{\xi(t)}\big(z(t), u(t)\big),\end{aligned}$$

where $\xi(k+1)$ is a discrete state, $\sigma(\cdot)$ governs the switching between the discrete states (refer to as "modes" in hybrid control), $\eta(\cdot)$ defines the transition function of the continuous state $x(k)$. Then the controlled platform can be depicted as the structure in Fig. 14.

The relation between the quantized DOB-net and the hybrid-system control can be found by comparing Figures 14 and 2. The components in the

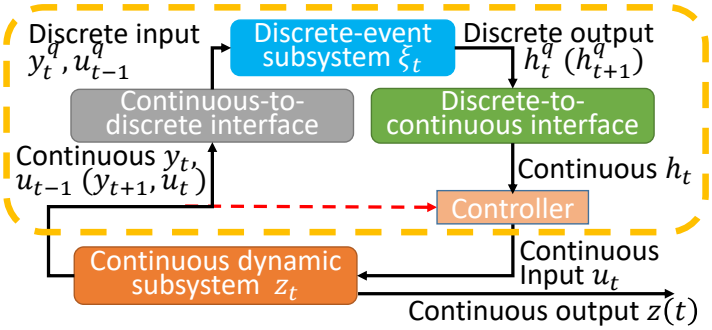


Figure 14: Hybrid structure of the controlled often-saturated system. Except for the red dashed arrow to the controller, components within the dash-rounded rectangle are equivalent to the DOB-net.

blue dashed rectangle in Fig. 2 correspond to the discrete-event subsystem, which is represented as the blue rounded rectangle in Fig. 14. The components in the green dash-dotted rectangle in Fig. 2 corresponds to the mapping between the discrete hidden states and the continuous controls, i.e., the discrete-to-continuous interface and the controller shown in Fig. 14.

Note that in classical hybrid modelling and control, the red dashed arrow to the controller is necessary, which is not kept in the quantized DOB-net. Therefore, the quantized DOB-net only captures the discrete-event subsystem, which partially describes the interplay between the control strategy and the environments. The DOB-net is able to estimate the discrete-event subsystem online and generate its sufficient representation for effective control.

Cyclic switchings were found in the learned DOB-net, showing the control policy is able to capture $\sigma(\cdot)$ for position regulation problems in different environments (different POMDPs). In Fig. 13, the control between switching was not depicted for clear illustration, which may reflect $\eta_\xi(\cdot)$. The continuous control based on feedback from continuous observation is missing in this study and should be included for future research.

8. CONCLUSION & FUTURE WORK

This paper proposes an attention-based abstraction approach for finding a key Moore machine network, which reveals the switching mechanism that has been captured in the DOB-net and is key to excessive disturbance rejection. This method is effective in abstracting control logic in solving

different POMDPs. Interestingly, the switching mechanism has been manually designed for controller developments in the existing literature. This finding may offer a bridge between DOB-nets and the hybrid systems for better network design. For example, in future we would design a special network/activation function to capture these saturation events and feed them into the control network, in addition to some continuous state representation.

The proposed A^2 approach is applicable to often-saturated systems. However, due to the current choice of sufficient attention, this approach may not be effective to non-saturated systems. In the future, more effort will be devoted to a new definition of sufficient attention to better capture the control mechanisms common in solving multiple POMDPs. For example, the choice of attention could be designed according to principle component analysis.

A^2 does not provide continuous control counterpart, therefore, the quantized DOB-net can not reach the performance of the continuous DOB-net. The continuous nature of the system control requires some complementary continuous controllers. In future, the continuous controls should be characterized to show how the system is guided between switchings, for the purpose of fully understanding the control network in the language of hybrid control. Another interesting future work is to investigate the possibility of using the switching mechanism obtained through inductive learning as some distilled knowledge for transfer learning.

References

[1] Abed-Alguni, B.H., Paul, D.J., Chalup, S.K., Henskens, F.A., 2016. A comparison study of cooperative q-learning algorithms for independent learners. Int. J. Artif. Intell 14, 71–93.

[2] Åström, K.J., Wittenmark, B., 2013. Adaptive control. Courier Corporation.

[3] Barnsley, M.F., 2014. Fractals everywhere. Academic press.

[4] Barreto, A., et al., 2017. Successor features for transfer in reinforcement learning, in: Advances in neural information processing systems, pp. 4055–4065.

[5] Bengio, Y., Léonard, N., Courville, A., 2013. Estimating or propagating gradients through stochastic neurons for conditional computation. arXiv preprint arXiv:1308.3432 .

[6] Benzaouia, A., Akhrif, O., Saydy, L., 2010. Stabilisation and control synthesis of switching systems subject to actuator saturation. International Journal of Systems Science 41, 397–409.

[7] Brahmbhatt, S., Hays, J., 2017. Deepnav: Learning to navigate large cities, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., pp. 3087–3096.

[8] Branicky, M., Borkar, V., Mitter, S., 1998. A unified framework for hybrid control: model and optimal control theory. IEEE Transactions on Automatic Control 43, 31–45.

[9] Camacho, E.F., Alba, C.B., 2013. Model predictive control. Springer Science & Business Media.

[10] Chen, W.H., Ballance, D.J., Gawthrop, P.J., O'Reilly, J., 2000. A nonlinear disturbance observer for robotic manipulators. IEEE Transactions on industrial Electronics 47, 932–938.

[11] Cheng, K.T., Krishnakumar, A.S., 1993. Automatic functional test generation using the extended finite state machine model, in: 30th ACM/IEEE Design Automation Conference, IEEE. pp. 86–91.

[12] Cleeremans, A., Servan-Schreiber, D., McClelland, J.L., 1989. Finite state automata and simple recurrent networks. Neural computation 1, 372–381.

[13] Crutchfield, J.P., Young, K., 1988. Computation at the onset of chaos, in: The Santa Fe Institute, Westview, Citeseer.

[14] Dong, C., Hou, Y., Zhang, Y., Wang, Q., 2010. Model reference adaptive switching control of a linearized hypersonic flight vehicle model with actuator saturation. Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering 224, 289–303.

[15] Doyle, J.C., Glover, K., Khargonekar, P.P., Francis, B.A., 1989. State-space solutions to standard h/sub 2/and h/sub infinity/control problems. IEEE Transactions on Automatic control 34, 831–847.

[16] Edwards, C., Spurgeon, S., 1998. Sliding mode control: theory and applications. Crc Press.

[17] Fierro, R., Lewis, F., 1997. A framework for hybrid control design. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans 27, 765–773.

[18] Finn, C., Abbeel, P., Levine, S., 2017. Model-agnostic meta-learning for fast adaptation of deep networks. arXiv preprint arXiv:1703.03400 .

[19] Fischer, N., Hughes, D., Walters, P., Schwartz, E.M., Dixon, W.E., 2014. Nonlinear rise-based control of an autonomous underwater vehicle. IEEE Transactions on Robotics 30, 845–852.

[20] Frasconi, P., Gori, M., Maggini, M., Soda, G., 1996. Representation of finite state automata in recurrent radial basis function networks. Machine Learning 23, 5–32.

[21] Gao, H., Cai, Y., 2016. Nonlinear disturbance observer-based model predictive control for a generic hypersonic vehicle. Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering 230, 3–12.

[22] Gao, Z., Huang, Y., Han, J., 2001. An alternative paradigm for control system design, in: Decision and Control, 2001. Proceedings of the 40th IEEE Conference on, IEEE. pp. 4578–4585.

[23] Ghafarirad, H., Rezaei, S.M., Zareinejad, M., Sarhan, A.A., 2014. Disturbance rejection-based robust control for micropositioning of piezoelectric actuators. Comptes Rendus Mécanique 342, 32–45.

[24] Gu, S., Lillicrap, T., Ghahramani, Z., Turner, R.E., Levine, S., 2016a. Q-prop: Sample-efficient policy gradient with an off-policy critic. arXiv preprint arXiv:1611.02247 .

[25] Gu, S., Lillicrap, T., Sutskever, I., Levine, S., 2016b. Continuous deep q-learning with model-based acceleration, in: International Conference on Machine Learning, pp. 2829–2838.

[26] Gunning, D., 2017. Explainable artificial intelligence (xai). Defense Advanced Research Projects Agency (DARPA), nd Web 2.

[27] Han, J., 1995. The" extended state observer" of a class of uncertain systems [j]. Control and Decision 1.

[28] Hinton, G.E., Salakhutdinov, R.R., 2006. Reducing the dimensionality of data with neural networks. science 313, 504–507.

[29] Hubara, I., Courbariaux, M., Soudry, D., El-Yaniv, R., Bengio, Y., 2016. Binarized neural networks, in: Advances in neural information processing systems, pp. 4107–4115.

[30] Johnson, C., 1968. Optimal control of the linear regulator with constant disturbances. IEEE Transactions on Automatic Control 13, 416–421.

[31] Karkus, P., Hsu, D., Lee, W.S., 2018. Particle filter networks: End-to-end probabilistic localization from visual observations. arXiv preprint arXiv:1805.08975 .

[32] Karpathy, A., Johnson, J., Fei-Fei, L., 2015. Visualizing and understanding recurrent networks. arXiv preprint arXiv:1506.02078 .

[33] Koul, A., Greydanus, S., Fern, A., 2018. Learning finite state representations of recurrent policy networks. arXiv preprint arXiv:1811.12530 .

[34] Lillicrap, T.P., Hunt, J.J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., Wierstra, D., 2015. Continuous control with deep reinforcement learning. arXiv preprint arXiv:1509.02971 .

[35] Lu, W., Liu, D., 2017. Active task design in adaptive control of redundant robotic systems, in: Australasian Conference on Robotics and Automation, ARAA.

[36] Lu, W., Liu, D., 2018. A frequency-limited adaptive controller for underwater vehicle-manipulator systems under large wave disturbances, in: The World Congress on Intelligent Control and Automation.

[37] Lu, W., Zhu, P., Ferrari, S., 2015. A hybrid-adaptive dynamic programming approach for the model-free control of nonlinear switched systems. IEEE Transactions on Automatic Control 61, 3203–3208.

[38] Lu, W., Zhu, P., Ferrari, S., 2016. An approximate dynamic programming approach for model-free control of switched systems. IEEE Transactions on Automatic Control .

[39] Mnih, V., Badia, A.P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., Kavukcuoglu, K., 2016. Asynchronous methods for deep reinforcement learning, in: International conference on machine learning, pp. 1928–1937.

[40] 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., 2015. Human-level control through deep reinforcement learning. Nature 518, 529.

[41] Nagabandi, A., Kahn, G., Fearing, R.S., Levine, S., 2018. Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning, in: Robotics and Automation (ICRA), 2018 IEEE International Conference on, IEEE. pp. 7579–7586.

[42] Oh, J., Chockalingam, V., Singh, S., Lee, H., 2016. Control of memory, active perception, and action in minecraft. arXiv preprint arXiv:1605.09128 .

[43] Ohishi, K., Nakao, M., Ohnishi, K., Miyachi, K., 1987. Microprocessor-controlled dc motor for load-insensitive position servo system. IEEE Transactions on Industrial Electronics , 44–49.

[44] Omlin, C.W., Giles, C.L., 1992. Training second-order recurrent neural networks using hints, in: Machine Learning Proceedings 1992. Elsevier, pp. 361–366.

[45] Paull, M.C., Unger, S.H., 1959. Minimizing the number of states in incompletely specified sequential switching functions. IRE Transactions on Electronic Computers , 356–367.

[46] Read, C., 2011. BP and the Macondo spill: the complete story. Springer.

[47] Sæmundsson, S., Hofmann, K., Deisenroth, M.P., 2018. Meta reinforcement learning with latent variable gaussian processes. arXiv preprint arXiv:1803.07551 .

[48] Samek, W., Wiegand, T., Müller, K.R., 2017. Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. arXiv preprint arXiv:1708.08296 .

[49] Schulman, J., Levine, S., Abbeel, P., Jordan, M., Moritz, P., 2015a. Trust region policy optimization, in: International Conference on Machine Learning, pp. 1889–1897.

[50] Schulman, J., Moritz, P., Levine, S., Jordan, M., Abbeel, P., 2015b. High-dimensional continuous control using generalized advantage estimation. arXiv preprint arXiv:1506.02438 .

[51] Skogestad, S., Postlethwaite, I., 2007. Multivariable feedback control: analysis and design. volume 2. Wiley New York.

[52] Sun, Z., Ge, S., 2005. Switched Linear Systems: Control and Design. Communications and Control Engineering, Springer.

[53] Umeno, T., Kaneko, T., Hori, Y., 1993. Robust servosystem design with two degrees of freedom and its application to novel motion control of robot manipulators. IEEE Transactions on Industrial Electronics 40, 473–485.

[54] Wang, S., Na, J., Ren, X., Yu, H., Yu, J., 2018. Unknown input observer-based robust adaptive funnel motion control for nonlinear servomechanisms. International Journal of Robust and Nonlinear Control 28, 6163–6179.

[55] Wang, S., Yu, H., Yu, J., 2019a. Robust adaptive tracking control for servo mechanisms with continuous friction compensation. Control Engineering Practice 87, 76–82.

[56] Wang, T., Lu, W., Yan, Z., Liu, D., 2019b. Dob-net: Actively rejecting unknown excessive time-varying disturbances. arXiv preprint arXiv:1907.04514 .

[57] Waslander, S., Wang, C., 2009. Wind disturbance estimation and rejection for quadrotor position control, in: AIAA Infotech@ Aerospace Conference and AIAA Unmanned... Unlimited Conference, p. 1983.

[58] Weiss, G., Goldberg, Y., Yahav, E., 2017. Extracting automata from recurrent neural networks using queries and counterexamples. arXiv preprint arXiv:1711.09576 .

[59] Woolfrey, J., Liu, D., Carmichael, M., 2016. Kinematic control of an autonomous underwater vehicle-manipulator system (auvms) using autoregressive prediction of vehicle motion and model predictive control, in: Robotics and Automation (ICRA), 2016 IEEE International Conference on, IEEE. pp. 4591–4596.

[60] Yang, J., Li, S., Chen, X., Li, Q., 2010. Disturbance rejection of ball mill grinding circuits using dob and mpc. Powder Technology 198, 219–228.

[61] Yosinski, J., Clune, J., Nguyen, A., Fuchs, T., Lipson, H., 2015. Understanding neural networks through deep visualization. arXiv preprint arXiv:1506.06579 .

[62] Yuan, C., Wu, F., 2015. Switching control of linear systems subject to asymmetric actuator saturation. International Journal of Control 88, 204–215.

[63] Zavlanos, M., Pappas, G., 2007. Distributed hybrid control for multiple-pursuer multiple-evader games, in: Hybrid Systems: Computation and Control. volume 4416 of *Lecture Notes in Computer Science*, pp. 787–789.

[64] Zoph, B., Le, Q.V., 2016. Neural architecture search with reinforcement learning. arXiv preprint arXiv:1611.01578 .

[65] Zuo, Z., Ho, D.W., Wang, Y., 2010. Fault tolerant control for singular systems with actuator saturation and nonlinear perturbation. Automatica 46, 569–576.