Journal of *Marine Science and Engineering*

MDPI

Article

Markovian-Jump Reinforcement Learning for Autonomous Underwater Vehicles under Disturbances with Abrupt Changes

Wenjie Lu [1,*], Yongquan Huang [1] and Manman Hu [2,*]

[1] School of Mechanical Engineering and Automation, Harbin Institute of Technology (Shenzhen), Shenzhen 518055, China
[2] Department of Civil Engineering, University of Hong Kong, Hong Kong, China
* Correspondence: luwenjie@hit.edu.cn (W.L.); mmhu@hku.hk (M.H.); Tel.: +86-15168551455 (W.L.)

Abstract: This paper studies the position regulation problems of an Autonomous Underwater Vehicle (AUV) subject to external disturbances that may have abrupt variations due to some events, e.g., water flow hitting nearby underwater structures. The disturbing forces may frequently exceed the actuator capacities, necessitating a constrained optimization of control inputs over a future time horizon. However, the AUV dynamics and the parameters of the disturbance models are unknown. Estimating the Markovian processes of the disturbances is challenging since it is entangled with uncertainties from AUV dynamics. As opposed to a single-Markovian description, this paper formulates the disturbed AUV as an unknown Markovian-Jump Linear System (MJLS) by augmenting the AUV state with the unknown disturbance state. Based on an observer network and an embedded solver, this paper proposes a reinforcement learning approach, Disturbance-Attenuation-net (MDA–net), for attenuating Markovian-jump disturbances and stabilizing the disturbed AUV. MDA–net is trained based on the sensitivity analysis of the optimality conditions and is able to estimate the disturbance and its transition dynamics based on observations of AUV states and control inputs online. Extensive numerical simulations of position regulation problems and preliminary experiments in a tank testbed have shown that the proposed MDA–net outperforms the existing DOB–net and a classical approach, Robust Integral of Sign of Error (RISE).

Keywords: autonomous underwater vehicles; disturbance rejection; reinforcement learning; markovian-jump systems

Citation: Lu, W.; Huang, Y.; Hu, M. Markovian-Jump Reinforcement Learning for Autonomous Underwater Vehicles under Disturbances with Abrupt Changes. *J. Mar. Sci. Eng.* **2023**, *11*, 285. https://doi.org/10.3390/jmse11020285

Academic Editor: Alessandro Ridolfi

Received: 19 December 2022
Revised: 18 January 2023
Accepted: 19 January 2023
Published: 27 January 2023

1. Introduction

Compared to Remotely Operated Vehicles (ROVs), Autonomous Underwater Vehicles (AUVs) may respond faster based on feedback from some perception modules or positioning systems and can thus enhance their performance in tasks, e.g., exploration, surveillance, cleaning bridge piles, and placing a heavy cover on a leaking oil well [1,2]. However, AUVs in shallow waters are often disturbed by inevitable strong disturbances. This research studies the control problem of stabilizing a control-input-saturated AUV under unknown excessive external disturbances [3–5]. Due to some events, the dynamics of external disturbances involve abrupt variations, making the transient performance of AUV unsatisfactory.

Much effort has been devoted to rejecting or attenuating disturbances in control problems since the 1980s. In particular, robust control [6], H-infinity control [7], adaptive control [8–10], and high-order sliding mode control [11] have been explored and used in industrial applications. In [12,13], Disturbance OBservers (DOBs) are used to estimate the lumped effects of unknown disturbances and uncertain dynamics models based on state observations and control inputs. From the first appearance of DOBs, many advanced ones have been studied, such as high-order disturbance observers for time series expansion and nonlinear systems [14,15].

Many control methods based on DOBs have since been developed, among which is the disturbance accommodation control [16–18]. Mismatched disturbance has been studied in the continuous and finite time regulation problem in [19], and values and multi-order derivatives of the disturbances are estimated to augment and stabilize the system via Lyapunov stability theorems.

However, the above-mentioned improvements in feedback controllers might fail to guarantee stability when the controlled system is subject to control saturation [20]. Small gain theorem might be explored in this case. However, it requires sufficiently accurate dynamics models, which are difficult to obtain for AUVs subject to various disturbances. These DOB-based approaches are effective when the disturbances are bounded and sufficiently small (compared to the actuator capacities) [21]. When the disturbance forces acting on the AUV frequently exceed the thrusters' capacities, the AUV can not be easily stabilized [3,22].

An ideal controller has to consider the (even saturated) controls' long-term effects on attenuating future disturbances; therefore, it is better to optimize the performance over a future time horizon, leading to constrained optimal control problems. Model Predictive Control (MPC) can deal with control input saturation and is thus an ideal candidate [20,23]. However, MPC usually requires a sufficiently accurate prediction model of the system [24], which might be unavailable due to the existence of disturbances in dynamics. Continuous-time MPC with a disturbance observer was proposed for disturbed systems in [25], where the disturbance estimations are utilized to adjust the prediction of the system output online, and an accurate AUV model is required. However, the latter is difficult to obtain, and, in addition, the disturbances are functions of time, which are unknown and difficult to measure via sensors. Another challenge associated with MPC is the computational burden of solving constrained optimization problems in real time at each time step.

The unknown dynamics models of AUV and the disturbances inhibit MPC approaches. These limitations of MPC and DOB-based controllers have led to DOB-nets [22], enabled by recent advances in Reinforcement Learning (RL). Model-free RL is adopted in this study since it does not require an explicit system model and can naturally adapt to noises and uncertainties [26]. Most existing RL methods build the controllers and critics in the domain of the AUV state space (the pose-velocity space). As a result, such RL can only capture the dynamics by the mappings between AUV state spaces, while the disturbances are functions of time and can not be described by mappings between AUV state spaces [27]. The unmodeled dynamics are treated as noises, which are further assumed to be independent and identically distributed (i.i.d.). Following this formulation, the noises are quite large and make the AUV system unstable, as shown in [22].

The key to this issue is to find an appropriate domain to define the dynamics model of the disturbed AUV and, thus, the controller, leaving the remaining unmodeled effects as noises of small moments. Many existing works model disturbances as superpositions of many harmonic oscillations [28]. In [22], model-free RL has been applied to reject excessive disturbances by modeling the control problems of the disturbed AUV via a set of unknown Partially Observable Markovian Decision Processes (POMDPs). The transition function (i.e., disturbed AUV dynamics) of each POMDP is heavily affected by disturbances. The input domain of the controller is built on the AUV state and the encoding of disturbance estimations in a future time horizon.

However, the work in [22] assumes that each POMDP has fixed harmonic oscillations, ignoring the fact that the disturbance dynamics may have abrupt changes. The dynamics of external disturbances are subject to abrupt changes due to events, e.g., a large vessel passing by, strong currents hitting underwater structures or oil well eruptions. Therefore, AUV's transient performance regarding these abrupt changes is unsatisfactory.

In this paper, it is shown that disturbances with varying characteristics can be modeled as a Markovian Jump Linear System (MJLS), the study of which has drawn a lot of attention. MJLS is found in many practical applications, and it aims to describe abrupt system variations caused by environmental changes. The disturbances are modeled by MJLS for

multiple disturbances in [29], where the transition matrix is partially known. A disturbance attenuation controller is constructed to achieve asymptotically stable performance. Singular Markovian-jump systems have been investigated in [30], where infinitely unobservable states are treated as unknown inputs. In addition, the approaches in the existing literature (e.g., [31,32]) do not consider control saturation, which widely exists and becomes an issue when the controlled objects are subject to excessive disturbances. To the best of the authors' knowledge, the stabilization problem of a control-input-saturated system subject to completely unknown and Markovian-jump disturbances has not been addressed.

Contribution: This study proposes a new RL approach referred to as MDA–net, which consists of a disturbance-dynamics-characteristics observer network (referred to as observer network) and an optimal controller network (referred to as controller network). Compared to DOB–net, MDA–net has three improvements.

(i) Different from the one in DOB-nets, the new observer network aims to learn the characteristics (i.e., frequencies, phases, and amplitudes) and their transition dynamics (i.e., properties describing Markovian-jump characteristics). The goal of this observer network is to provide the feature description of the in situ disturbed AUV system dynamics to the control network.

(ii) A two-step learning approach (module learning and end-to-end learning) is adopted, which is regularized by the process of the disturbance prediction built on the disturbance harmonic model (the superposition of multiple disturbances). The observer network outputs a feature representation of the quadratic optimization problem in the encoding space, which is further referred to as the problem feature in the remainder of this paper. It is natural to train a solver (a controller network) that receives these problem features and outputs control signals. However, it is difficult to learn a solver of optimization problems purely from data. A Quadratic Programming (GP) solver is embedded in the controller network.

(iii) In this paper, the gradients of the optimization over the problem features are established based on the sensitivity analysis of optimization regarding the QP solver and are then used to train the controller network together with the critics.

In the remainder of this paper, the formulation of the position regulations problems is given in Section 2, and then the previous work DOB–net is reviewed in Section 3. After that, Sections 4 and 5 present the MJLS formulation and the proposed MDA–net, respectively. Section 6 summarizes the implementation details and the results from the numerical simulation and experiments in lab conditions. The limitations and potential improvements are discussed in Section 7, followed by conclusions in Section 8.

2. Problem Formulation

The position regulation problem arises from many underwater applications. The stabilization of AUVs is particularly important to inspection or intervention tasks where the AUV platform is free-floating and affected by the disturbance. AUV platforms often have sufficiently large restoring forces and can thus be kept horizontal. Therefore, the pitch and roll motions are not considered. The desired restoring forces can be achieved by designing the distance between the buoyancy center and the mass center. The surge, sway, heave, and yaw motions of the AUV platform are heavily affected by the excessive disturbances and thus require a proper controller.

Let $\mathbf{q} \in \mathbb{R}^3 \times SO(2)$ denote the AUV position and heading, and the AUV velocities and accelerations $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$ are in the tangent space $\mathbb{R}^4$ of the manifold. It is assumed that $\mathbf{q}$ and $\dot{\mathbf{q}}$ are obtained from the perception system, e.g., Simultaneous Localization And Mapping (SLAM). In clean water with steady illumination, cameras can be used, while in other cases, an onboard multi-beam sonar can be used, as reported in [33]. The dynamics of the disturbed AUV are given as

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

where $\mathbf{M} \in \mathbb{R}^{4\times4}$ denotes the inertia matrix, $\mathbf{C} \in \mathbb{R}^{4\times4}$ denotes the matrix of the Coriolis and centripetal terms, $\mathbf{D} \in \mathbb{R}^{4\times4}$ denotes the matrix of the drag force, and $\mathbf{g} \in \mathbb{R}^4$ denotes the vector of the lumped gravity-buoyancy forces. As pointed out in [34], it is quite difficult to measure these terms, which depend on the flow density and velocities. Therefore, in this study, these matrices and vectors are unknown to the controller.

In the studied problems, the disturbances are represented by their equivalent forces acting on the AUV platform. In the remainder of this paper, "disturbances" and "disturbance forces" are used exchangeably, and they are denoted by $\mathbf{d} \in \mathbb{R}^4$. The control $\mathbf{u} \in \mathcal{U}$ is saturated at bounds $\bar{\mathbf{u}} = \max(\mathcal{U}) \in \mathbb{R}^4$ and $\underline{\mathbf{u}} = \min(\mathcal{U}) \in \mathbb{R}^4$, where max and min are dimension-wise operators, and $\mathcal{U} \subset \mathbb{R}^4$ is a compact set of control. For simplicity, the bounds on each dimension of $\mathbf{u}$ are independent of each other. This assumption might not be true if the total power from all thrusters is restricted by the AUV's power supply.

Definition 1 (Excessive External Disturbances)**.** *The disturbances are called excessive if their forces* $\mathbf{d}$ *acting on the AUV frequently exceed the control saturation* $\bar{\mathbf{u}}$ *and* $\underline{\mathbf{u}}$*.*

The external disturbances in this study are excessive to the actuators' capabilities (see Definition 1). Definition 1 only makes sense if the control inputs and the disturbance forces enter the AUV system from the same channel. In other cases, a similar definition might be explored by mapping the control inputs and disturbances into the same channel. In a real AUV system, the disturbance forces may enter the system from a different channel as control inputs $\mathbf{u}$. The experimental results have shown the formulation in Equation (1) is reasonable.

Problem 1 (Optimal Control Problem)**.** *Obtain a controller that outputs actions* $\mathbf{u}$ *to the system (1), such that an objective function is maximized in an episode under disturbances of randomly generated and abruptly changed characteristics. System (1) is discretized in time, and the objective function is defined as the discounted sum of the collected rewards,*

$$J = \sum_{\tau=0}^{T-1} \gamma^\tau r(\mathbf{x}_\tau), \tag{2}$$

where $r(\boldsymbol{x}_t) \triangleq -\mathbf{x}_t^T R \mathbf{x}_t$*,* $\boldsymbol{x}_t$ *is the AUV state at time* t*,* T *denotes the number of time steps in an episode, and* $\gamma \in [0, 1)$ *is a discount factor that prioritizes the near-term rewards [35].*

The optimization of Equation (2) is subject to Equation (1) and control saturations $\bar{\mathbf{u}}$ and $\underline{\mathbf{u}}$. More importantly, the obtained controller should be applicable to Problem 1 with various randomly generated and abruptly changed disturbances.

3. Previous Work: DOB–net

The control problems of the heavily disturbed AUV cannot be precisely described by a single POMDP in the AUV state space. Augmented state spaces have been studied to better describe the dynamics of the disturbed AUV. Based on the assumption that recent states and actions together encode the transition functions of a POMDP at the visited states, a history-window control approach has been developed, and it takes in as inputs a number of most recent states and actions [36]. Similarly, the disturbed AUV dynamic system has been modeled as a multi-order Markovian chain in [37]. However, it might be difficult to determine the number of orders. A small number of orders might not rediscover the POMDP characteristics, while a large number of orders make the training and generalization of the trained policy challenging.

The DOB–net approach, proposed in [22], is built on the classical actor-critic architecture, as described in Figure 1. DOB–net utilizes hidden states from Gated Recurrent Units (GRUs) to encode the transition function of the multi-order Markovian chain. DOB–net consists of an observer network and a controller network, as shown in Figure 1. The

observer network is built upon GRUs to mimic the dynamics involved in DOBs and the dynamics of time series prediction. The controller network outputs the control signals and critic values, as required byA2C. Since the controller is also a function of the AUV state, state **x** is aggregated with the hidden state from the observer network.

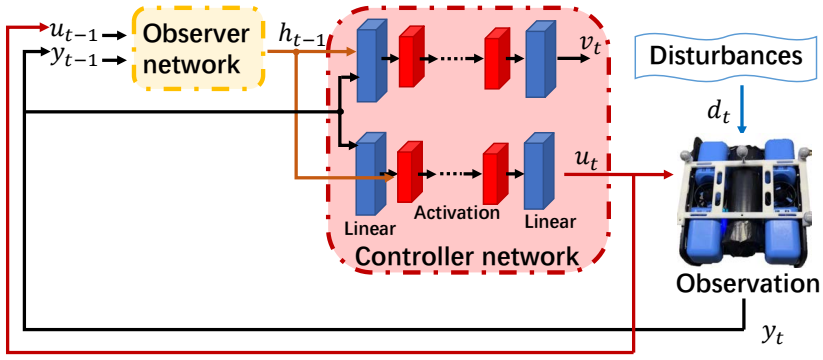


Figure 1. Network architecture of DOB–net.

The DOB–net (the observer network and the controller network) is trained in an end-to-end manner through interactions with the disturbed AUV systems [38–40]. The procedures of training and testing contain a number of episodes, where each episode contains T time steps. DOB–net outperforms the existing approach RISE. However, in [22], the disturbances considered have constant characteristics in each episode. The abrupt changes in these disturbances are not considered. As a result, the transient performance in abrupt events is poor.

4. Markovian Jump Linear System

This section shows that by modeling disturbances as the superpositions of multiple harmonic oscillations, Problem 1 can be modeled as a Markovian Jump Linear System (MJLS). With this modeling, it is reasonable to embed a QP solver in the controller network. This layer of the QP solver is different from regular layers (activation layers or hidden layers); it involves running QP to solve for solutions. The details of this QP layer are introduced in Section 5.

As pointed out in [41], the harmonic disturbance in the channel of the control input is given by the exogenous systems,

$$\begin{aligned} \mathbf{d}_t &= \mathbf{V}[\mathbf{s}_t]\boldsymbol{\omega}_t, \\ \boldsymbol{\omega}_{t+1} &= \boldsymbol{\omega}_t + \mathbf{W}[\mathbf{s}_t]\boldsymbol{\omega}_t + \mathbf{G}[\mathbf{s}_t]\mathbf{e}_t, \end{aligned} \tag{3}$$

where $\mathbf{s}_t$ is a discrete-time Markovian process, $\boldsymbol{\omega}$ is the internal state of the disturbance, and $\mathbf{e}_t$ is square integrable over time horizon $[0, \infty)$, i.e., $\mathbf{e}_t \in \mathcal{L}_2[0, \infty)$. The integration of each signal's $\mathcal{L}_2$ norm in the signal set $\mathcal{L}_2[0, \infty)$ is less than infinite. There is noise, $\mathbf{e}_t$, from the perturbations and uncertainties from the exogenous systems. Often, matrix $\mathbf{W}(\mathbf{s})$ has the following form with $c > 0$,

$$\begin{bmatrix} 0 & c \\ -c & 0 \end{bmatrix}, \tag{4}$$

where c is the frequency of the harmonic oscillation. Harmonic disturbances widely exist in many practical engineering problems, and the frequency is often assumed to be known while the phase and the amplitude are often estimated online. Many existing approaches are able to attenuate disturbances under this assumption [41]. However, in the studied problems, this assumption is invalid due to the complexity of the disturbed AUV dynamics. Moreover, based on the superposition assumption, the numbers in Equation (3) are aggregated to describe the disturbance considered in this paper.

The discrete-time Markovian process $\mathbf{s}_t \in \mathcal{S}$ is defined as follows. Then, let the matrix Pr denote the transition probability. The discrete-time systems $\{\mathbf{s}_t\}_{t=0,1,\cdots}$ is a time-homogeneous Markovian chain that takes values from a finite set $\mathcal{S} = \{\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_S\}$ with stationary transition probabilities.

$$p_{ij} = \Pr(\mathbf{s}_{t+1} = \mathbf{s}_j | \mathbf{s}_t = \mathbf{s}_i), \tag{5}$$

where $p_{ij} \geq 0$ is the transition probability from mode i at time t to mode j at time $t+1$ and $\sum_{j=1}^{M} p_{ij} = 1$. The abrupt changes are, in fact, represented by the transition probability matrix T. The cardinality of $\mathcal{S}$ and elements in $\mathcal{S}$ are implicitly learned from the data, as shown in Section 5.

Substituting Equation (3) into Equation (1) yields

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{C}\dot{\mathbf{q}} + \mathbf{D}\dot{\mathbf{q}} + \mathbf{g} = \mathbf{u} + \mathbf{V}[\mathbf{s}_t]\boldsymbol{\omega}_t. \tag{6}$$

In addition, **M**, **C**, and **D** are functions of $\dot{\mathbf{q}}$. In order to model the disturbed AUV dynamics model as a Markovian linear jump system, we have

$$\begin{aligned} \mathbf{M} &= \bar{\mathbf{M}} + \tilde{\mathbf{M}} \\ \mathbf{C} &= \bar{\mathbf{C}} + \tilde{\mathbf{C}} \\ \mathbf{D} &= \bar{\mathbf{D}} + \tilde{\mathbf{D}}, \end{aligned} \tag{7}$$

where $\bar{\mathbf{M}}$, $\bar{\mathbf{C}}$, $\bar{\mathbf{D}}$ are the dominant and fixed part of the matrices **M**, **C**, and **D**, respectively, while $\tilde{\mathbf{M}}$, $\tilde{\mathbf{C}}$, and $\tilde{\mathbf{D}}$ are the residuals and are subject to change. By converting Equation (6) into a discrete-time model, we have

$$\dot{\boldsymbol{q}}_{t+1} = \dot{\boldsymbol{q}}_t + (-\mathbf{M}^{-1}\mathbf{C}\dot{\mathbf{q}} - \mathbf{M}^{-1}\mathbf{D}\dot{\mathbf{q}} - \mathbf{M}^{-1}\mathbf{g} + \mathbf{M}^{-1}\mathbf{u} + \mathbf{M}^{-1}\mathbf{V}[\mathbf{s}_t]\boldsymbol{\omega}_t)dt$$

Let $\mathbf{z} \triangleq [\mathbf{q}^T, \dot{\mathbf{q}}^T, \boldsymbol{\omega}^T]^T$ denote the aggregated system state, Equations (3) and (8) together yield

$$\mathbf{z}(k+1) = \mathbf{A}[\mathbf{s}_t]\mathbf{z}(k) + \mathbf{B}\mathbf{u} + \mathbf{E} + \mathbf{H}[s_t]\mathbf{e} + \delta, \tag{8}$$

where

$$\mathbf{A}[\mathbf{s}_t] \triangleq \begin{bmatrix} \boldsymbol{I} & 0 & 0 \\ 0 & -\bar{\mathbf{M}}^{-1}(\bar{\mathbf{C}} + \bar{\mathbf{D}})dt & \bar{\mathbf{M}}^{-1}\mathbf{V}[\mathbf{s}_t]dt \\ 0 & 0 & \mathbf{W}[\mathbf{s}_t] \end{bmatrix}, \tag{9}$$

$$\mathbf{B} \triangleq [\mathbf{0}^T, \bar{\mathbf{M}}^{-1}dt, \mathbf{0}^T]^T, \tag{10}$$

$$\mathbf{E} \triangleq [\mathbf{0}^T, \mathbf{g}^T\bar{\mathbf{M}}^{-1}dt, \mathbf{0}^T]^T, \tag{11}$$

$$\mathbf{H}[\mathbf{s}_t] \triangleq [\mathbf{0}^T, \mathbf{0}^T, \mathbf{G}[\mathbf{s}_t]^T\mathbf{M}^{-1}dt]^T, \tag{12}$$

and

$$\delta \triangleq [-\tilde{\mathbf{M}}\ddot{\mathbf{q}} - \tilde{\mathbf{C}}\dot{\mathbf{q}} - \tilde{\mathbf{D}}\dot{\mathbf{q}}]dt \tag{13}$$

is the lumped uncertainties from the remaining unmodeled dynamics not captured by other terms. In this paper, δ is treated as Gaussian noise.

It is now shown that Problem 1 can be modeled as an MJLS. Thus, the aggregated space of the AUV state, the disturbance state, and the disturbance characteristics is a sufficient

input space for the controller network. This finding is used in Section 5 to design the MDA–net.

5. MDA–net

MDA–net consists of an observer network and a controller network. The observer network is designed to estimate the MJLS parameters, i.e., the encodings of **A**, **B**, **E**, and **D**. With the parameters of MJLS available; they are converted into an optimization problem, which is then solved by the QP layer. The obtained optimal "solution" is then mapped to control inputs to the AUV.

5.1. Observer Network

Based on the fact that the disturbances are harmonic, a new observer network is designed to learn the disturbance characteristics that vary according to a Markovian chain, as shown in Figure 2. The harmonic model is integrated into the MDA–net to mimic a disturbance prediction process. The connected modules in the observer network establish a pipeline from the estimation of the disturbance state (i.e., the hidden state h_{t-1}) to the inference of the disturbance characteristics f_{t-1}, and then the Markovian-jump properties g_{t-1}. Given the current state of disturbances and the parameters of the harmonic model, the expectation of future disturbances ($\hat{d}_t$) can be predicted.

The design of the observer network is based on the design in [41], where the disturbance state ω and the nonlinear term σ are unknown. In [41], the observer can be designed as,

$$\begin{aligned}\hat{\mathbf{d}}_t &= \mathbf{V}_i\hat{\omega}_t\\ \dot{\hat{\omega}}_t &= \mathbf{v}_t - \mathbf{L}_i\mathbf{x}_t\\ \dot{\mathbf{v}}_t &= (\mathbf{W}_i + \mathbf{L}_i\mathbf{G}_i\mathbf{V}_i)(\mathbf{v} - \mathbf{L}_i\mathbf{x}_t) + \mathbf{L}_i(\mathbf{A}_i\mathbf{x}_t + \mathbf{G}_i\mathbf{u}_t)\end{aligned} \tag{14}$$

where, for notational simplicity, $\mathbf{A}(\mathbf{s}_i)$ is denoted by $\mathbf{A}_i$, and $\mathbf{G}(\mathbf{s}_i)$ and $\mathbf{H}(\mathbf{s}_i)$ are denoted by $\mathbf{G}_i$ and $\mathbf{H}_i$, respectively. The observer works when the parameters $\mathbf{V}_i$, $\mathbf{W}_i$, $\mathbf{A}_i$, and $\mathbf{G}_i$ are known.

On the other hand, Gated Recurrent Units (GRUs) are similar to Long Short-Term Memory (LSTM) but with fewer parameters. A GRU is as follows,

$$\begin{aligned}z_t &= \sigma(W_z[h_{t-1}, \boldsymbol{x}_t, \boldsymbol{u}_t] + b_z)\\ r_t &= \sigma(W_r[h_{t-1}, \boldsymbol{x}_t, \boldsymbol{u}_t] + b_r)\\ \tilde{h}_t &= \tanh(W_h[r_t \circ h_{t-1}, \boldsymbol{x}_t, \boldsymbol{u}_t] + b_h)\\ h_t &= (1 - z_t) \circ h_{t-1} + z_t \circ \tilde{h}_t\end{aligned} \tag{15}$$

where $\boldsymbol{x}_t$ and $\boldsymbol{u}_t$ are the inputs, h_t is the output vector, z_t is the gate vector, r_t is the reset vector, W and b are the weight matrices and bias vectors, $\circ$ denotes the Hadamard product, and σ and $tanh$ are the activation functions (sigmoid function and hyperbolic tangent).

Based on the similarity between the observer in [41] and GRU, we propose the observer network, as shown in Figure 2. The network is able to offer more flexibility and can deal with superpositioned disturbances. Partially unknown transition probabilities are investigated in [31]. However, due to underwater environments, it may not be trivial to have the probability matrix available. Therefore, this paper studies the MJLSs with the finite set $\mathcal{S}$ and probability matrix T are completely unknown and are to be inferred from the data. In order to capture the fact that the parameters of the harmonic model could jump, GRU#2 takes $\hat{d}_t$ as inputs to estimate the Markovian jumps and reset the hidden state of GRU#1 and GRU#2 if abrupt events are detected by GRU#3.

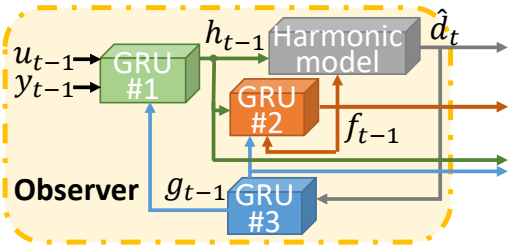


Figure 2. Observer network.

5.2. Controller Network

The outputs from the observer network f_{t-1}, h_{t-1}, and g_{t-1} are also fed into the controller network, together with the most recent observation y_t. When feeding f_{t-1} into the harmonic model, it is reshaped and separated to form matrices defined in Equation (9). The MJLS nature of the disturbed system permits the existence of optimization problems at each time step. The controller network should be able to solve the optimization problem. The controller network consists of a transform network module and a QP solver SNOPT, as shown in Figure 3.

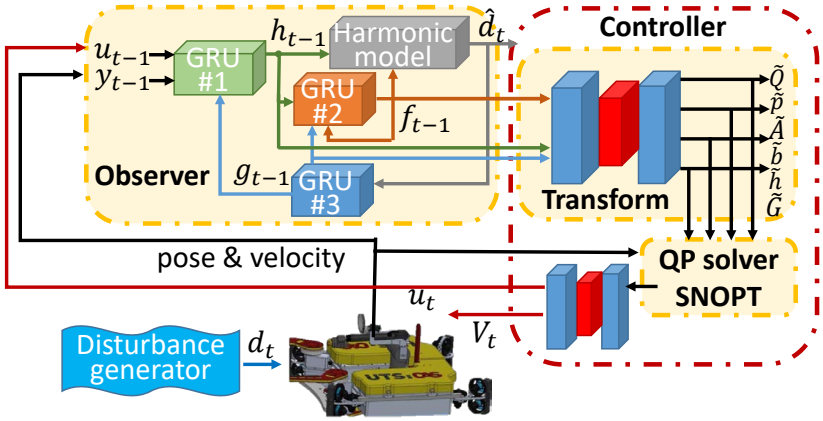


Figure 3. Network Architecture of MDA–net.

It is worth pointing out that the superpositions of disturbances make the formulation of the optimization untrivial. Therefore, optimization problems are not manually designed but are learned from the learning process. MDA–net consists of a transform module to convert the parameters of MJLS into the parameters of the constrained quadratic optimization problems.

Let $\xi \triangleq [z_0^T, \cdots, z_{T-1}^T, u_0^T, \cdots, u_{T-1}^T]^T$, Equation (2) can be rewritten as the following convex Quadratic Programming (QP) problem,

$$\xi^* = \arg\min_{\xi} \frac{1}{2}\xi^T \tilde{Q}\xi + \tilde{p}^T\xi, \tag{16}$$

subject to

$$\tilde{A}\xi = \tilde{b}, \text{ and, } \tilde{G}\xi \leq \tilde{h}, \tag{17}$$

where $\tilde{Q}$, $\tilde{A}$, $\tilde{p}$, and $\tilde{b}$ are transformed from the hidden states f, h, and g, while $\tilde{G}$ and $\tilde{h}$ are constraints.

The QP layer then takes in as inputs the parameters of the constrained quadratic optimization problems. The constrained convex optimization problem can be solved by many existing QP tools, such as SNOPT by Gill [42]. The outputs of the QP solver are with respect to the problem encoded by the hidden layers. A second transform module is used

to map the outputs of the solver to the control and critic. The critic is used to train the system in an RL fashion. The implementation details can be found in Section 6.

5.3. Network Training

It is found that the end-to-end training approach failed to train the MDA–net. Therefore, the supervised learning of the observer network is conducted first, and then the entire network is trained in Advantage Actor Critic (A2C) fashion [43].

5.3.1. Observer Network Training

GRUs #1 and #2 are designed to estimate the encodings of the disturbances and the parameters of the harmonic model, respectively. The learning process is regularized by the harmonic model and the known disturbances during training. The prediction procedure is described as follows. Given the disturbance transition function and the previous state of disturbances, Equation (3) is used iteratively N times to produce N-step predictions. The third GRU #3 detects possible Markovian jumps, and it enforces the hidden state in GRUs #1 and #2. GRU #3 is able to improve the estimates of the encodings of the disturbances and the parameters of the harmonic model at transient instants.

The observer network is trained by the targeted disturbance values $d_t, d_{t+1}, \cdots, d_{t+N-1}$ at the N future steps, which are available during training (not available during testing), as shown in Algorithm 1. The trainable parameters in the observer network are denoted as θ_o. Then the loss function used in this supervised learning at time t in each episode is given as

$$\mathcal{L}_1 = \frac{1}{N} \sum_{1 \le \tau \le N} \| \hat{d}_{t-1+\tau} - d_{t-1+\tau} \|, \tag{18}$$

where $\| \cdot \|$ denotes the mean square error and $t \le T - N + 1$. The integration of the harmonic model enforces the hidden state f_{t-1} and h_{t-1} to lie in the disturbance parameter space and the disturbance state space, respectively. Then the Markovian-jump properties can be estimated by GRU #3.

K-step rollouts of GRUs #1 and #2 are conducted when applying the truncated Back Propagation Through Time (truncated BPTT). The observer network is then trained by using a Stochastic Gradient Descent (SDG) approach [44]. The data set is obtained by numerical simulations, and the control inputs are generated by a PID controller. Each sample in this data set consists of y_{t-1}, u_{t-1}, $\{d_\tau\}_{t \le \tau \le N}$. When the disturbed system becomes unstable, the simulations are reset with randomly generated initial conditions.

Algorithm 1: Observer-network.

Data acquisition: $\{y_{t_0}, u_{t_0}, d_{t_0}, \cdots, y_t, u_t, d_t\}_i$.
Truncation parameter: K.
Truncation index k reset.
for $j \in \{t_0, \ldots, t-p\}$ **do**
  if *episode ends* **then**
    Hidden states h_{t-1} and f_{t-1} reset.
  end
  if $k == K$ **then**
    Truncation index k reset.
    Hidden states h_{t-1} and f_{t-1} detach from computation graph.
  end
  Gradient accumulation: $d\theta_o \leftarrow d\theta_o + \partial \mathcal{L}_1(j) / \partial \theta_o$.
  k=k+1.
end
Synchronous update : θ_o using $d\theta_o$.

5.3.2. Controller Network Training

While f_t might be statistically sufficient to describe the optimal control problem corresponding to the encountered disturbances, it may require a large network with many hidden layers and a large amount of data to learn an optimal controller network to deal with all possible disturbances. In this paper, a solver is added to the controller network. The training of the controller network is to find a suitable presentation of the optimization problem ($\tilde{\boldsymbol{Q}}$, $\tilde{\boldsymbol{A}}$, $\tilde{\boldsymbol{p}}$, and $\tilde{\boldsymbol{b}}$) of the estimates of the MJLS.

Recall that the optimization problem has the following form,

$$\xi^* = \arg\min_{\xi} \frac{1}{2}\xi^T \tilde{Q}\xi + \tilde{p}^T\xi, \tag{19}$$

subject to

$$\tilde{A}\xi = \tilde{b}, \text{ and, } \tilde{G}\xi \leq \tilde{h}. \tag{20}$$

Since the number of harmonic components in Equation (3) is difficult to know a priori, therefore, a transform network module is used to encode it into a suitable feature space to enhance the flexibility of the trained MDA–net.

Without loss of generality, active inequality constraints with regard to the current solution is denoted as $\tilde{G}\xi = \tilde{h}$, then the above optimal problem is represented by a linear optimization problem with the equality constraints, as shown in [45],

$$\begin{bmatrix} \tilde{Q} & \tilde{A}^T & \tilde{G}^T \\ \tilde{A} & 0 & 0 \\ \tilde{G} & 0 & 0 \end{bmatrix} \begin{bmatrix} \xi^* \\ \lambda^* \\ \nu^* \end{bmatrix} = - \begin{bmatrix} \tilde{p} \\ \tilde{b} \\ \tilde{h} \end{bmatrix}, \tag{21}$$

where λ^* and ν^* are the Lagrange multipliers.

In order to use the A2C framework and the backpropagation technique to train the transform network, the sensitivity analysis of optimization problems is applied in order to provide more informative updates to learn the controller network. In other words, the updates have to go through the solver SNOPT. SNOPT is an iterative method, and it is possible to roll out its iterative steps, and backpropagate gradients are possible, which are, however, slow. Another approach is to consider the derivatives of the optimal control regarding the problem parameters ($\tilde{\boldsymbol{Q}}$, $\tilde{\boldsymbol{A}}$, $\tilde{\boldsymbol{p}}$, and $\tilde{\boldsymbol{b}}$).

Then, the derivatives of ξ^* with respect to $\tilde{A}$ are obtained by

$$\nabla_{\tilde{A}}\xi^*(l) = d_\lambda^* \otimes \xi^*(l) + \lambda^* \otimes d_\xi^*(l), \tag{22}$$

where $\otimes$ is an element-wise operator, $\xi^*(l)$ is the lth entry of ξ^*, and d_ξ^* and d_λ^* are the solutions of the following linear system,

$$\begin{bmatrix} \tilde{Q} & \tilde{A}^T & \tilde{G}^T \\ \tilde{A} & 0 & 0 \\ \tilde{G} & 0 & 0 \end{bmatrix} \begin{bmatrix} d_\xi^* \\ d_\lambda^* \\ d_\nu^* \end{bmatrix} = - \begin{bmatrix} \nabla_{\xi^*}\xi^*(l) \\ 0 \\ 0 \end{bmatrix}. \tag{23}$$

The gradient $\nabla_{\tilde{A}}\xi^*(l)$ is, in fact, what the controller network should offer during backpropagation training since the controller network is designed to behave as a constrained linear quadratic optimal problem solver. Therefore, the gradient $\nabla_{\tilde{A}}\xi^*(l)$ is used to train the controller network, along with critics (value functions), within the A2C framework.

Running multiple environment instances across threads, A2C utilizes synchronous gradient descents to learn the controller network, leading to statistically stationary critics and gradients. Furthermore, the gradients are averaged over multi-step updates. The parameter updates by A2C are given as

$$\nabla_{\theta'} \log \pi\left(u_t | x_t, h_{t-1}, f_{t-1}, g_{t-1}; \theta'\right) A(x_t, u_t, h_{t-1}, f_{t-1}, g_{t-1}; \theta, \theta_v),$$

where $A(x_t, u_t, h_{t-1}, f_{t-1}, g_{t-1}; \theta, \theta_v)$ is an estimate of the advantage function given by

$$\sum_{i=0}^{k-1} \gamma^i r_{t+i} + \gamma^k V(x_{t+k}, u_{t+k-1}, h_{t+k-1}; \theta_v) - V(x_t, h_{t-1}, f_{t-1}, g_{t-1}; \theta_v), \tag{24}$$

where $k \leq N$ can vary from state to state. The purpose of the baseline $V(x_t, h_{t-1}, f_{t-1}, g_{t-1}; \theta_v)$ is to have a smaller variance of the advantage function values and, thus, the gradients. Note that PyTorch might not allow backpropagating through in-place-modified variables. The issue is from the "zero_grad" function in Pytorch, and it can be worked around by manually zeroing the gradients, as shown in Algorithm 2.

Algorithm 2: A2C for a thread T.

Initialize globally shared parameters θ_u and θ_v.
Initialize thread-related parameters $\theta_u^{\mathcal{T}}$ and $\theta_v^{\mathcal{T}}$.
repeat
  if *episode ends* **then**
    $t = 0$, Sample new environment.
  end
  $d\theta_u = 0$, $d\theta_v = 0$, $\theta_u^{\mathcal{T}} = \theta_u$, and $\theta_v^{\mathcal{T}} = \theta_v$.
  Obtain y_t, h_{t-1}, f_{t-1}, g_{t-1}, and f_{t-1}.
  repeat
    Update control, hidden state, reward, state:
    $u_t, h_t = \pi(u_t | x_t, h_{t-1}, f_{t-1}, g_{t-1}; \theta_u^i)$.
    $r_{t+1}, z_{t+1} = f_e(z_t, u_t)$, $y_{t+1} = o(z_{t+1})$.
    $t \leftarrow t + 1$.
  until *episode ends or truncated*;
  $$A = \begin{cases} 0 & \text{episode ends} \\ V(y_t, h_{t-1}, f_{t-1}, g_{t-1}; \theta_v^{\mathcal{T}}) & \text{otherwis} \end{cases}.$$
  for $j \in \{t-1, \ldots, t+M\}$ **do**
    $A \leftarrow r_j + \gamma A$
    Update gradient: $d\theta_u^{\mathcal{T}} \leftarrow d\theta_u^{\mathcal{T}} + \nabla_{\theta_u^{\mathcal{T}}} \log \pi(u_j | y_j, u_{j-1}, h_{j-1}; \theta_u^{\mathcal{T}}) \cdot (A - V(y_j, u_{j-1}, h_{j-1}; \theta_v^{\mathcal{T}}))$
    Accumulate gradient: $d\theta_v^{\mathcal{T}} \leftarrow d\theta_v^{\mathcal{T}} + \partial (A - V(y_j, u_{j-1}, h_{j-1}; \theta_v^{\mathcal{T}}))^2 / \partial \theta_v^{\mathcal{T}}$
  end
  For all $\mathcal{T}$, update θ_u by $d\theta_u^{\mathcal{T}}$ and θ_v by $d\theta_v^{\mathcal{T}}$.
until *Converge*;

6. Implemetion and Simulations

This section first describes a simulated position regulation problem arising from field applications, such as the remediation of a spewing well. Different from the scenario used in DOB–net [22], the changes in the disturbance characteristics are simulated, including abrupt changes. Then, an MDA–net with the hand-picked structure parameters is introduced, as well as the hyperparameters used in training, followed by simulation results.

6.1. Position Regulation

The AUV characteristics, such as mass, control capabilities, disturbance amplitudes, etc., are proportionally scaled. The AUV mass was set to 1 (kg), and the control saturation was set to $\bar{u} = -\underline{u} = [2,\ 2,\ 2]^T\ (N)$. As discussed in Section 2, the translational and yaw motions of the AUV platform are considered. The simulated external disturbances are three-dimensional but do not act through the center of mass of the platform to introduce

disturbances in heading control. The disturbance force in each axis is harmonic and their superposition is given as

$$d_t = \begin{bmatrix} A^x \sin(\frac{\pi}{T^x}t + \phi^x) \\ A^y \sin(\frac{\pi}{T^y}t + \phi^y) \\ A^z \sin(\frac{\pi}{T^z}t + \phi^z) \end{bmatrix}, \tag{25}$$

where the parameters at time 0 in each episode is given by

$$\begin{aligned} A^x(0), A^y(0), A^z(0) &\sim U(1,3) \\ T^x(0), T^y(0), T^z(0) &\sim U(2,4) \\ \phi^x(0), \phi^y(0), \phi^z(0) &\sim U(-\pi, \pi), \end{aligned} \tag{26}$$

and $U(a,b)$ is the uniform distribution over the interval $[a,\ b]$.

These parameters of the disturbances change according to the following Markovian chain, where the change rate ρ at each time step is set to 0.1. When abrupt changes occur, the variations are sampled as follows,

$$\begin{aligned} \delta A_t^x, \delta A_t^y, \delta A_t^z &\sim \mathcal{N}(0,1) \\ \delta T_t^x, \delta T_t^y, \delta T_t^z &\sim \mathcal{N}(0,1) \\ \delta \phi_t^x, \delta \phi_t^y, \delta \phi_t^z &\sim \mathcal{N}(0,1), \end{aligned} \tag{27}$$

where $\mathcal{N}$ is a normal distribution, the sampled values are added onto the current disturbance parameters, the results of which are then saturated by the ranges given in Equation (26). When training the algorithm, in each episode, a small noise (about 5 percent) was added to the value of the AUV mass. It is because the added mass is a function of the velocity and geometry of the AUV; this noise can make the controller robust. However, when the mass changes a lot, the algorithm performs poorly, and retraining must be conducted using more accurate estimations of the mass.

6.2. MDA–net Implementation

The structural parameters of the proposed MDA–net are summarized in Table 1, where each GRU only has one recurrent layer. The learning rate in training the observer network was set to 1×10^{-4}, and the training took about 7 h on a 2.5 GHz Intel i5 CPU. Both learning rates for MDA–net and DOB–net were set to 7×10^{-4}, with 16 threads running simultaneously. Their training took about 4.2 and 3.7 h, respectively.

Table 1. Network structure parameters.

GRU Index	**#1**	**#2**	**#3**
hidden neurons	32	32	32
Layer of transform #1	#1	#2	#3
(input, output)	(1024, 512)	(256, 128)	(128,128)
Layer of transform #2	#1	#2	#3
(input, output)	(32, 32)	(32, 16)	(16,4)

6.3. Prediction Performance

An example from numerical simulations showing the performance of the disturbance prediction when an abrupt change occurs is given in Figure 4. The sudden change was sampled from Equation (26). Multi-step prediction (2.5 (s) into the future) is performed. As observed in Wang's work [22], model predictive control over a time horizon of 2.5 (s) is sufficient for underwater robots under the disturbances described by Equation (26). The solid curves show the ground truth, and the dashed curves with markers illustrate the

predictions from the observer network. Notice that the dashed curve segment with the same and consecutive markers illustrates one 2.5 (s) prediction. This example is showing five such predictions. The abrupt change occurred around the seventh second, and the prediction immediately became worse. However, the observer network in MDA–net is able to quickly infer the changed disturbance characteristics. While the DOB–net cannot deal with disturbances with Markovian jumps effectively, as shown in Figure 5.

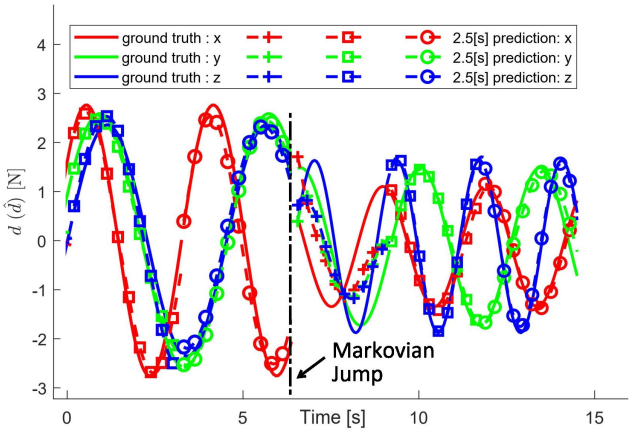


Figure 4. Disturbance prediction example by MDA–net. Solid curves showing ground truth; marked curves showing 2.5 (s) prediction from three different instants.

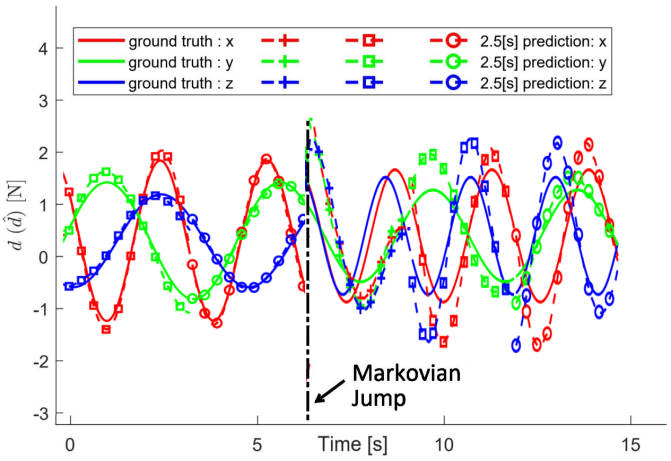


Figure 5. Disturbance prediction example by DOB–net. Solid curves showing ground truth; marked curves showing 2.5 (s) prediction from three different instants.

6.4. Stabilization Performance

In training and testing the controller network, the frequencies, amplitudes, and phrases of the disturbances are all randomly generated. This section compares the performances of DOB–net and the proposed MDA–net in solving Problem 1. The training score of the DOB–net is averaged over episodes and is about -713.3, and the averaged score of the MDA–net is about -455.6. Both scores are calculated according to Equation (2). The difference in training scores has shown that the MDA–net outperforms the DOB–net in dealing with the Markovian-jump disturbances simulated in this paper. As reported in [22], when dealing with disturbances whose characteristics do not change in an episode, the training score of the DOB–net could reach about -200.

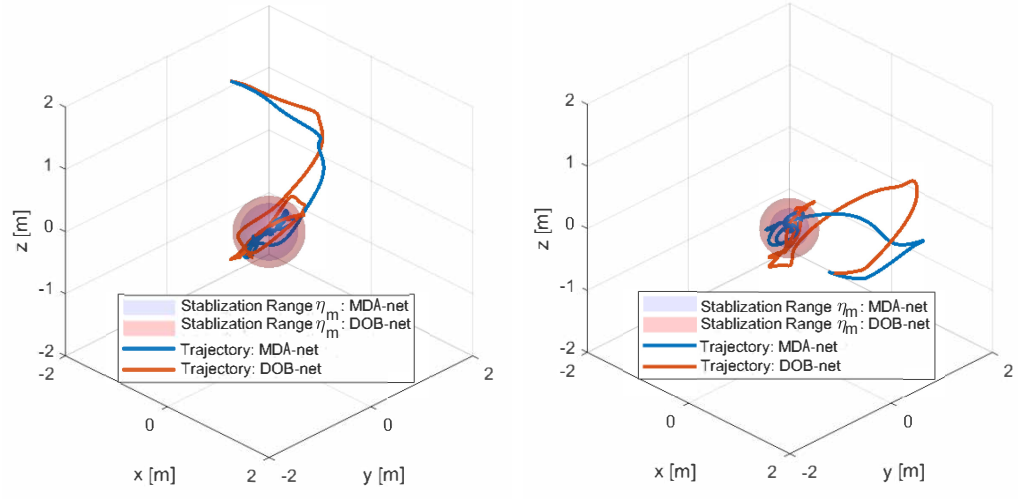
Since the goal of the platform stabilization is to reach a minimum stabilization range, the regulation error is defined as the distance between the AUV platform to the targeted position (assumed the origin in the inertial space), given as

$$\eta = \|q\|. \tag{28}$$

Then the stabilization range is defined as the largest error after 5 seconds in an episode, as follows,

$$\eta_m = \max_{5<t} \eta_t. \tag{29}$$

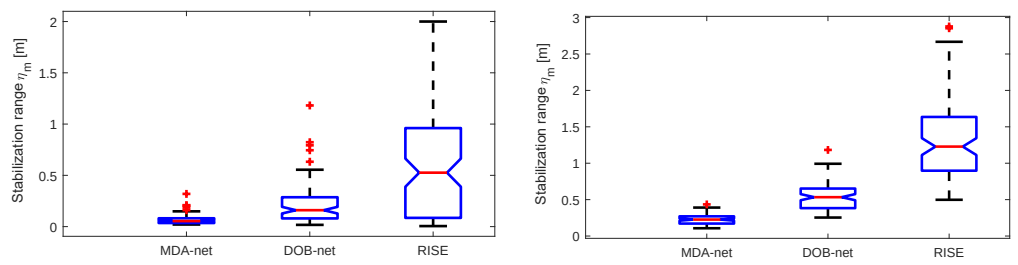
Two trajectory examples are illustrated in Figure 6a and 6b, respectively. In the first example, the disturbance amplitude exceeds the control saturation by 10 percent, while in the second example, the disturbance amplitudes are 80 percent of the control saturation. In both cases, the changes in the disturbances were given by Equation (26). The transparent spheres in blue and the ones in red indicate the stabilization ranges obtained by the MDA–net and the DOB–net, respectively. Both examples show that the proposed MDA–net has a smaller stabilization range than the DOB–net [22] in rejecting excessive disturbances subject to abrupt changes. The trajectories obtained from RISE were not shown for a clear illustration; the obtained stabilization ranges were often quite large. The smaller range indicates less challenge for the onboard manipulators, which is not discussed in this paper. The limitations of these numerical simulations are discussed in the last section.



(**a**) Trajectory example #1 (**b**) Trajectory example #2

Figure 6. Examples of the trajectories obtained by MDA–net and DOB–net, respectively. (**a**) The stabilization range eta_m obtained from MDA–net is 0.43 (m) and eta_m obtained from DOB–net is 0.54 (m); (**b**) The stabilization range eta_m obtained from MDA–net is 0.22 (m), and eta_m obtained from DOB–net is 0.45 (m).

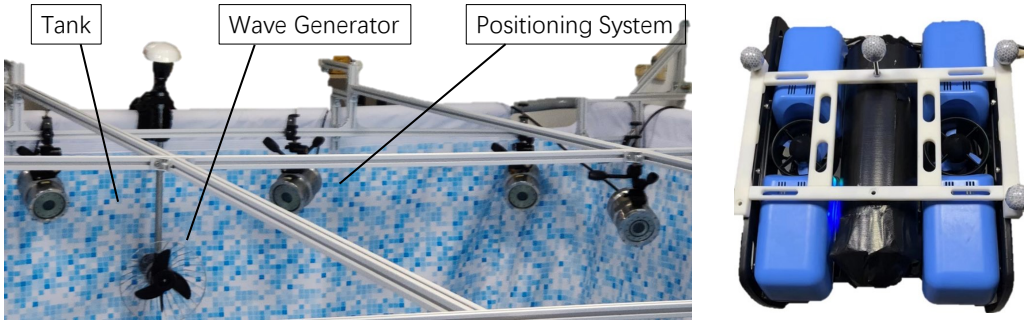
We have also conducted extensive comparisons between MDA–net, DOB–net, and RISE in two different groups of scenarios. The amplitudes of the simulated disturbances in Group #1 could exceed the control saturation levels by 10%, while the amplitudes of the simulated disturbances in Group #2 could exceed the control saturation levels by 30% RISE controller [46]. In Figure 7a and Figure 7b, the results have shown that the MDA–net outperformed DOB–net and RISE in the test cases.



(**a**) Comparison in Group #1. (**b**) Comparison in Group #2.

Figure 7. Comparison among MDA–net, DOB–net, and RISE under disturbances in Groups #1 and #2.

MDA–net was also tested in a tank, where the water flow was generated by a propeller fixed on the edge of the tank, as shown in Figure 8a. The direction of this propeller can be manually adjusted to create various disturbances. In the experiments, the sudden external impact was from the sudden changes in the disturbance generated by the position-fixed propeller. Its direction oscillated through manual control. The strength of the propeller force was adjusted as follows. By connecting the AUV to the frame fixed on the tank via a force-torque sensor, the forces acted on the AUV by disturbances were measured. The PWM signals to the propeller motor were adjusted such that the external forces acting on the AUV reached in [20, 40] N.



(**a**) Tank setup. (**b**) AUV with four markers.

Figure 8. Testbed description: tank, wave generator, positioning system, and AUV with markers.

The saturation of the control inputs was confined to 30 N by setting the maximum value of the PWM signals of AUV thrusters. The AUV mass was about 14.5 kg. When testing the controller, AUV was detached from the force-torque sensor. Therefore, an example of the disturbance generated by the position-fixed propeller is given in Figure 9. The sudden changes were simulated by changing the propeller force and direction abruptly.

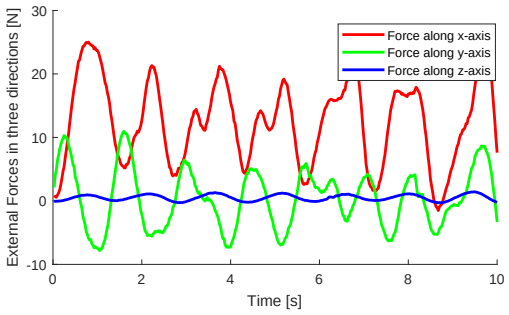
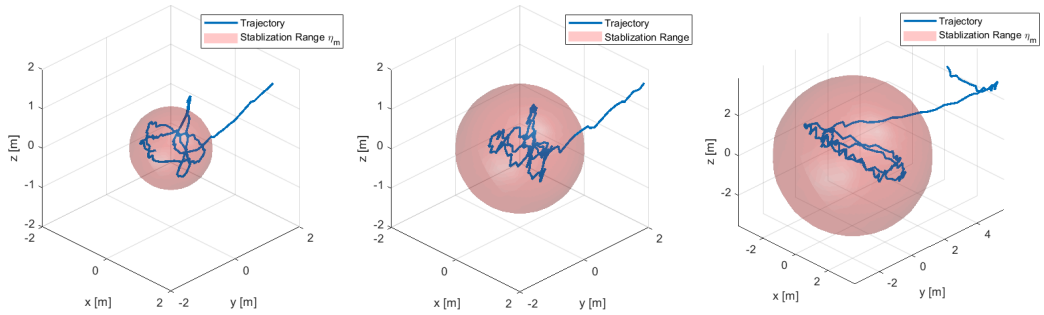


Figure 9. The forces in the $x-$ and $y-$directions are shown in red and green, respectively. The disturbance in the $z-$direction (shown in blue) is negligible.

An underwater positioning system was implemented with 12 cameras that emit blue lights, as illustrated in Figure 8a. The AUV has four highly reflective markers on top Figure 8b. In addition, due to the low visibility underwater, the reflective markers are 30 mm wide. The positioning system was calibrated with an L-shaped bar with markers of known body coordinates. The cameras capture the markers and outputs pose estimates at 60 Hz.

The AUV system was built on BlueRov2 from Blue Robotics, Inc., Torrance, CA, USA as shown in Figure 8b. BlueRov2 is equipped with six thrusters and can translate in three directions, roll, and yaw. BlueRov2 communicates with a desktop and receives thruster commands at 50 Hz. The desktop also receives the pose estimates from the positioning system. In this AUV testbed, the desktop implements the DMA-net, making the BlueRov2 an AUV. The limitation of this setup is discussed in Section 7.

The obtained AUV trajectories are shown in Figure 10. The three approaches, MDA–net, DOB–net, and the RISE controller, were compared. The regulation errors and stabilization ranges of which are also shown in Figure 11. The limitation of the localization system in the lab tank will be discussed in Section 7.



(**a**) MDA–net (**b**) DOB–net (**c**) RISE

Figure 10. An example of trajectory and position regulation errors from tank tests. The blue curves represent the trajectories from three approaches, and the transparent spheres present stabilization ranges η_m defined in Equation (29).

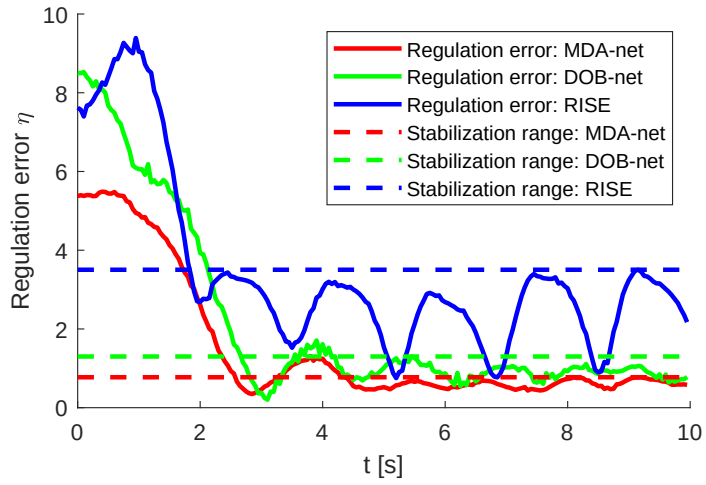


Figure 11. Regulation errors η (solid curves) and stabilization ranges η_m (dashed lines) obtained from MDA–net (in red), DOB–net (in green), and RISE (in blue), respectively.

One example of these experimental tests was shown in Figure 10; the AUVs started from the position around $[1.1, 1.8, 1.2]^T$. Three trajectories were separately obtained by three approaches, and they are shown in Figure 10a, 10b, and 10c, respectively. The trajectories have demonstrated that the MDA–net approach can offer better performance with a smaller stabilization range, while RISE can hardly keep the AUV near the origin.

7. Discussion and Future Work

The training of the controller does involve a lot of computational loads. When the trained controller is used online, at each time step, the computational cost is $O(n_n n_l)$, where n_n is the number of neurons in each layer and n_l is the number of layers, while the RISE approach is about $O(1)$. As observed, the evaluations of a trained controller network on a low-voltage CPU can achieve 100 Hz, which is sufficient for online applications.

Since the existence of the transform module, it is difficult to interpret the learned representation of the QP problem. It is, therefore, hard to analyze the stability of the system. In the future, additional regulations from Lyapunov's theories should be imposed on the learning of controller networks.

While using disturbance knowledge in training might be avoided by closing the supervision from the aggregation of the platform model simulator and regularization on

the simulator outputs, the platform model is also a module in the network, which could be extracted from GRU #1.

In addition, the abrupt changes in disturbances are naively made, and they may not reflect actual field situations. Future work includes testing the proposed approach in more realistic underwater environments. In addition, future improvements include orthogonal learning for different environments since MJLS and the sensitivity analysis have already provided a framework to extract principle components when a deep network is employed.

The limitation of the tank tests arises from the positioning approach since, in the real world, an auxiliary camera system is not available, and the sonar positioning system is often too noisy and has low bandwidth. In the future, an onboard underwater multi-band sonar sensor, camera, and localization system will be investigated.

The laptop receives the pose estimation from the motion capture system, implements the proposed RL approach, and sends the control signals to the ROV. The whole system is referred to as the testbed of the "AUV". The testbed is not equivalent to AUV systems since the cameras are mounted along tank edges. However, the testbed may be sufficient to test the proposed control algorithm. We are developing an underwater sonar SLAM system to provide online state estimation to make the underwater fully autonomous. In the future, the sonar-based localization approach and the proposed transfer RL algorithm will be implemented in the updated hardware of the ROV, making it a real AUV.

8. Conclusions

This paper proposes an RL approach, referred to as MDA–net, for stabilizing a free-floating platform subject to excessive harmonic disturbances and control saturation. Through modeling the disturbed AUV platform as an MJLS, the harmonic model is integrated into the network for effective learning of the observer of the MJLS parameters. Sensitivity analysis of the optimal control problems is used to guide the learning of the controller network. Preliminary results from numerical simulations and tank tests have shown that MDA–net outperforms DOB–net when the disturbances have abrupt changes.

Author Contributions: Conceptualization, W.L. and M.H.; methodology, W.L. and M.H.; software, W.L. and Y.H.; validation, W.L. and Y.H.; resources, W.L.; writing—original draft preparation, W.L.; writing—review and editing, Y.H.; visualization, W.L.; supervision, W.L.; funding acquisition, W.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the National Natural Science Foundation of China #62003110 and the Shenzhen Science and Technology Innovation Foundation #JCYJ20210324132607018, #JSGG20210420091804012, and #GXWD20220811163649003.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data is available at https://www.wenjielu.cn accessed on 18 January 2023.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Griffiths, G. *Technology and Applications of Autonomous Underwater Vehicles*; CRC Press: Boca Raton, FL, USA, 2002; Volume 2.
2. Woolfrey, J.; Lu, W.; Liu, D. A Control Method for Joint Torque Minimization of Redundant Manipulators Handling Large External Forces. *J. Intell. Robot. Syst.* **2019**, *96*, 3–16. [CrossRef]
3. Xie, L.L.; Guo, L. How much uncertainty can be dealt with by feedback? *IEEE Trans. Autom. Control* **2000**, *45*, 2203–2217.
4. Gao, Z. On the centrality of disturbance rejection in automatic control. *ISA Trans.* **2014**, *53*, 850–857. [CrossRef] [PubMed]
5. Li, S.; Yang, J.; Chen, W.H.; Chen, X. *Disturbance Observer-Based Control: Methods and Applications*; CRC press: Boca Raton, FL, USA, 2014.
6. Skogestad, S.; Postlethwaite, I. *Multivariable Feedback Control: Analysis and Design*; Wiley: New York, NY, USA, 2007; Volume 2.
7. Doyle, J.C.; Glover, K.; Khargonekar, P.P.; Francis, B.A. State-space solutions to standard H/sub 2/and H/sub infinity/control problems. *IEEE Trans. Autom. Control* **1989**, *34*, 831–847. [CrossRef]
8. Åström, K.J.; Wittenmark, B. *Adaptive Control*; Courier Corporation: Washington, DC, USA, 2013.

9. Lu, W.; Liu, D. Active task design in adaptive control of redundant robotic systems. In Proceedings of the Australasian Conference on Robotics and Automation (ARAA 2017), Sydney, Australia, 11–13 December 2017.
10. Lu, W.; Liu, D. A frequency-limited adaptive controller for underwater vehicle-manipulator systems under large wave disturbances. In Proceedings of the World Congress on Intelligent Control and Automation, Changsha China, 4–8 July 2018.
11. Salgado-Jimenez, T.; Spiewak, J.M.; Fraisse, P.; Jouvencel, B. A robust control algorithm for AUV: Based on a high order sliding mode. In Proceedings of the OCEANS'04 MTTS/IEEE TECHNO-OCEAN'04, Kobe, Japan, 9–12 November 2004; Volume 1, pp. 276–281.
12. Chen, W.H.; Ballance, D.J.; Gawthrop, P.J.; O'Reilly, J. A nonlinear disturbance observer for robotic manipulators. *IEEE Trans. Ind. Electron.* **2000**, *47*, 932–938. [CrossRef]
13. Chen, W.H.; Ballance, D.J.; Gawthrop, P.J.; Gribble, J.J.; O'Reilly, J. Nonlinear PID predictive controller. *IEE Proc.-Control Theory Appl.* **1999**, *146*, 603–611. [CrossRef]
14. Kim, K.S.; Rew, K.H.; Kim, S. Disturbance observer for estimating higher order disturbances in time series expansion. *IEEE Trans. Autom. Control* **2010**, *55*, 1905–1911.
15. Su, J.; Chen, W.H.; Li, B. High order disturbance observer design for linear and nonlinear systems. In Proceedings of the 2015 IEEE International Conference on Information and Automation, Beijing, China, 2–5 August 2015; pp. 1893–1898.
16. Johnson, C. Optimal control of the linear regulator with constant disturbances. *IEEE Trans. Autom. Control* **1968**, *13*, 416–421. [CrossRef]
17. Johnson, C. Accomodation of external disturbances in linear regulator and servomechanism problems. *IEEE Trans. Autom. Control* **1971**, *16*, 635–644. [CrossRef]
18. Chen, W.H.; Yang, J.; Guo, L.; Li, S. Disturbance-observer-based control and related methods—An overview. *IEEE Trans. Ind. Electron.* **2015**, *63*, 1083–1095. [CrossRef]
19. Li, S.; Sun, H.; Yang, J.; Yu, X. Continuous finite-time output regulation for disturbed systems under mismatching condition. *IEEE Trans. Autom. Control* **2014**, *60*, 277–282. [CrossRef]
20. Gao, H.; Cai, Y. Nonlinear disturbance observer-based model predictive control for a generic hypersonic vehicle. *Proc. Inst. Mech. Eng. Part I J. Syst. Control Eng.* **2016**, *230*, 3–12. [CrossRef]
21. Ghafarirad, H.; Rezaei, S.M.; Zareinejad, M.; Sarhan, A.A. Disturbance rejection-based robust control for micropositioning of piezoelectric actuators. *Comptes Rendus Mécanique* **2014**, *342*, 32–45. [CrossRef]
22. Wang, T.; Lu, W.; Yan, Z.; Liu, D. DOB–net: Actively rejecting unknown excessive time-varying disturbances. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020; pp. 1881–1887.
23. Camacho, E.F.; Alba, C.B. *Model Predictive Control*; Springer Science & Business Media: Berlin, Germany, 2013.
24. Maeder, U.; Morari, M. Offset-free reference tracking with model predictive control. *Automatica* **2010**, *46*, 1469–1476. [CrossRef]
25. Yang, J.; Zheng, W.X.; Li, S.; Wu, B.; Cheng, M. Design of a prediction-accuracy-enhanced continuous-time MPC for disturbed systems via a disturbance observer. *IEEE Trans. Ind. Electron.* **2015**, *62*, 5807–5816. [CrossRef]
26. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; MIT Press: Cambridge, MA, USA, 2018.
27. Sæmundsson, S.; Hofmann, K.; Deisenroth, M.P. Meta reinforcement learning with latent variable gaussian processes. *arXiv* **2018**, arXiv:1803.07551.
28. Kormushev, P.; Caldwell, D.G. Improving the energy efficiency of autonomous underwater vehicles by learning to model disturbances. In Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, 3–7 November 2013; pp. 3885–3892.
29. Sun, H.; Li, Y.; Zong, G.; Hou, L. Disturbance attenuation and rejection for stochastic Markovian jump system with partially known transition probabilities. *Automatica* **2018**, *89*, 349–357. [CrossRef]
30. Yao, X.; Park, J.H.; Wu, L.; Guo, L. Disturbance-observer-based composite hierarchical antidisturbance control for singular Markovian jump systems. *IEEE Trans. Autom. Control* **2018**, *64*, 2875–2882. [CrossRef]
31. Zhang, L.; Boukas, E.K. Stability and stabilization of Markovian jump linear systems with partly unknown transition probabilities. *Automatica* **2009**, *45*, 463–468. [CrossRef]
32. Zhang, J.; Shi, P.; Lin, W. Extended sliding mode observer based control for Markovian jump linear systems with disturbances. *Automatica* **2016**, *70*, 140–147. [CrossRef]
33. Rahman, S.; Li, A.Q.; Rekleitis, I. Svin2: An underwater slam system using sonar, visual, inertial, and depth sensor. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; pp. 1861–1868.
34. Antonelli, G. *Underwater Robots*; Springer: Cham, Switzerland, 2014; Volume 3.
35. Nagabandi, A.; Kahn, G.; Fearing, R.S.; Levine, S. Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 21–25 May 2018; pp. 7579–7586.
36. Sandholm, T.W.; Crites, R.H. Multiagent reinforcement learning in the iterated prisoner's dilemma. *Biosystems* **1996**, *37*, 147–166. [CrossRef] [PubMed]
37. Wang, T.; Lu, W.; Liu, D. Excessive Disturbance Rejection Control of Autonomous Underwater Vehicle using Reinforcement Learning. In Proceedings of the Australasian Conference on Robotics and Automation 2018, Lincoln, New Zealand, 4–6 December 2018.

38. van der Himst, O.; Lanillos, P. Deep Active Inference for Partially Observable MDPs. *arXiv* **2020**, arXiv:2009.03622.
39. Hausknecht, M.; Stone, P. On-policy vs. off-policy updates for deep reinforcement learning. In Proceedings of the Deep Reinforcement Learning: Frontiers and Challenges, IJCAI 2016 Workshop, New York, NY, USA, 9–11 July 2016.
40. Oh, J.; Chockalingam, V.; Singh, S.; Lee, H. Control of memory, active perception, and action in minecraft. *arXiv* **2016**, arXiv:1605.09128.
41. Yao, X.; Guo, L. Composite anti-disturbance control for Markovian jump nonlinear systems via disturbance observer. *Automatica* **2013**, *49*, 2538–2545. [CrossRef]
42. Gill, P.E.; Murray, W.; Saunders, M.A. SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM Rev.* **2005**, *47*, 99–131. [CrossRef]
43. Mnih, V.; Badia, A.P.; Mirza, M.; Graves, A.; Lillicrap, T.; Harley, T.; Silver, D.; Kavukcuoglu, K. Asynchronous methods for deep reinforcement learning. In Proceedings of the International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016; pp. 1928–1937.
44. Bottou, L. Large-scale machine learning with stochastic gradient descent. In Proceedings of the COMPSTAT'2010, Paris, France, 22–27 August 2010; Physica-Verlag: Heidelberg, Germany, 2010; pp. 177–186.
45. Amos, B.; Jimenez, I.; Sacks, J.; Boots, B.; Kolter, J.Z. Differentiable MPC for end-to-end planning and control. In Proceedings of the 2018 Conference on Neural Information Processing Systems, Montreal, QC, Canada, 3–8 December 2018; pp. 8289–8300.
46. Fischer, N.; Kan, Z.; Kamalapurkar, R.; Dixon, W.E. Saturated RISE feedback control for a class of second-order nonlinear systems. *IEEE Trans. Autom. Control* **2013**, *59*, 1094–1099. [CrossRef]