Learning Based Active Rejection of Environmental Disturbances for Underwater Robots

by

Tianming Wang

A thesis submitted in partial fulfilment of the
requirements for the degree of Doctor of Philosophy

at the
Centre for Autonomous Systems
Faculty of Engineering and Information Technology
University of Technology Sydney

June 2020

Certificate of Original Authorship

I, Tianming Wang declare that this thesis, is submitted in fulfillment of the requirements for the award of Doctor of Philosophy, in the Faculty of Engineering and Information Technology at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise reference or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program and the University of Technology Sydney.

Signed: Production Note:
Signature removed prior to publication.

Date: 02 Jun 2020

Learning Based Active Rejection of Environmental Disturbances for Underwater Robots

by

Tianming Wang

A thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy

Abstract

Underwater robots in shallow waters usually suffer from turbulent flows and strong waves. Such disturbances may frequently exceed the robot's control constraints, thus severely destabilize robot during task operation. Conventional disturbance observer and model predictive control are not particularly effective since they heavily rely on a sufficiently accurate dynamics model. Learning-based controllers are able to alleviate model dependency and achieve high computational efficiency. Learned control policies normally specialize on one dynamics model, and may not directly generalize to other models. Transfer learning offers a pathway to bridge the mismatch between different dynamics models. In this thesis, reinforcement learning algorithms are applied that enables optimal control of underwater robots under unobservable excessive time-correlated disturbances, and transfer learning algorithms are implemented for control policy adaptation under dynamics model mismatch.

History Window Reinforcement Learning (HWRL) and Disturbance Observer Network (DOB-Net) are developed for disturbance rejection control. Both algorithms jointly optimize a disturbance observer and a motion controller, and implicitly learn embedding of disturbance waveforms from motion history of robot. A modular design of learning disturbance rejection controller is also developed. A Generalized Control Policy (GCP) is trained over a wide range of disturbance waveforms, an Online Disturbance Identification

Model (ODI) exploits motion history of robot to predict the disturbance waveforms, which served as input to GCP. Together, GCP-ODI provides robust control across a wide variety of disturbances.

Transfer learning algorithms are applied to address the mismatch between a mathematical model of system dynamics developed from the fundamental principles of dynamics and an empirical model of system dynamics derived from real-world experimental data. Hybrid Policy Adaptation (HPA) is first proposed where learning a model-free policy under the empirical model is accelerated by pre-training a model-based policy with the mathematical model. Transition Mismatch Learning (TML) is then proposed that learns a compensatory policy based on the modular architecture of GCP-ODI through minimizing transition mismatch between the mathematical model and the empirical model.

Numerical simulations on a pose regulation task have demonstrated that HWRL, DOB-Net and GCP-ODI can successfully stabilize the underwater robot across a wide range of disturbance waveforms, and outperform conventional controllers and classical RL policies. Both HPA and TML achieve satisfactory control performance when deployed under the empirical model, with high sample efficiency and avoidance of initial exploratory actions.

Acknowledgements

First and foremost, I would like to express my deep gratitude to my principal supervisor Prof. Dikai Liu, for providing me with an opportunity to do a PhD and introducing me to the field of robotics. He gave me maximum freedom on research so that I can explore different and interesting topics. He always encouraged me to combine theory and practice, his enthusiasm and guidance inspired me to be a good robotics researcher.

I would also like to thank my co-supervisor Dr. Wenjie Lu for his continuous support in both research and life during my entire PhD journey. He always shared with me knowledge and experience in research methodologies and skills, and gave me many helpful feedbacks and suggestions on my projects and papers. He set an example for me on how to be a good researcher.

I would like to thank the excellent researchers I have met during my PhD study. Enormous thanks to Huan Yu, Jonathan Woolfrey, Teng Zhang, Karthick Thiyagarajan, Jiaheng Zhao, Yanhao Zhang, Yongbo Chen, Kanzhi Wu, Fang Bai for many inspiring discussions on research. I would like to thank the research engineers at the Centre for Autonomous Systems, Andrew To, Khoa Le, Thomas Hudson, Li Yang Liu and Brenton Leighton, for their generous help in experiments for my research. I would like to thank the postdoctoral researchers, Zheng Yan and Junyu Xuan, at the Centre for Artificial Intelligence, for the discussions and advice on machine learning.

Besides, I would like to thank my girlfriend for her endless patience listening to me talk about my research and encouragement whenever I got depressed.

Finally, I would like to thank my parents for their unconditional support, both financially and spiritually, throughout my whole student life.

Contents

Declaration of Authorship **iii**

Abstract **v**

Acknowledgements **vii**

List of Figures **xiii**

List of Tables **xvii**

1 Introduction **1**

1.1 Background 1

1.2 Motivation 3

1.3 Contributions 6

1.3.1 Learning Optimal Control under Excessive Time-Correlated Disturbances 6

1.3.2 Transferring Disturbance Rejection Policy under Dynamics Model Mismatch 8

1.4 Thesis Outline 10

1.5 Publications 11

2 Review of Related Work **13**

2.1 Anti-Disturbance Control 13

2.1.1 Disturbance Attenuation 13

2.1.2 Disturbance Rejection 14

2.2 Reinforcement Learning in Partially Observable Markov Decision Processes 15

2.2.1 Recurrent Neural Networks in Reinforcement Learning 15

2.2.2 Meta-Reinforcement Learning 17

2.3 Transfer Learning in Reinforcement Learning 18

2.3.1 Model-Based Reinforcement Learning 19

2.3.2 Domain Randomization 21

2.4 Summary 22

3 Joint Learning of Disturbance Rejection Control with History Embedding 25

3.1 Problem Formulation . . . 26

3.1.1 System Dynamics . . . 26

3.1.2 Control Objective . . . 28

3.1.3 Reinforcement Learning . . . 29

3.2 History Window Reinforcement Learning . . . 30

3.2.1 Validation of Multi-Order Markov Chain . . . 31

3.2.2 Learning History Window Policy . . . 32

3.3 Disturbance Observer Network . . . 35

3.3.1 Conventional Disturbance Observer . . . 35

3.3.2 Gated Recurrent Unit . . . 37

3.3.3 Learning Disturbance Observer Network . . . 37

3.4 Simulation . . . 40

3.4.1 Simulation Setup . . . 40

3.4.2 Results of History Window Reinforcement Learning . . . 44

3.4.3 Results of Disturbance Observer Network . . . 47

3.4.4 Results on Real-World Wave Data . . . 53

4 Modular Learning of Disturbance Rejection Control with Online Disturbance Identification 55

4.1 Learning a Generalized Control Policy with Online Disturbance Identification 55

4.1.1 Learning Generalized Control Policy . . . 57

4.1.2 Learning Online Disturbance Identification Model . . . 58

4.2 Simulation . . . 61

4.2.1 Simulation Setup . . . 61

4.2.2 Results . . . 62

5 Transferring Disturbance Rejection Policy with Hybrid Policy Adaptation 65

5.1 Problem Formulation . . . 66

5.2 Model-Based Policy Optimization . . . 67

5.3 Hybrid Policy Adaptation . . . 69

5.3.1 Learning Model-Free Compensatory Policy . . . 70

5.3.2 Model-Based Initialization of Model-Free Reinforcement Learning . . 71

5.4 Construction of Empirical Model . . . 72

5.4.1 Data Collection . . . 72

5.4.2 Kalman Filtering . . . 75

5.4.3 Dynamics Model Learning . . . 78

5.5 Simulation . . . 79

5.5.1 Results of Model-Based Policy Optimization . . . 79

5.5.2 Results of Hybrid Policy Adaptation . . . 80

6 Transferring Disturbance Rejection Policy with Transition Mismatch Compensation 85

6.1 Compensatory Control Learning . . . 86
6.2 Transition Mismatch Learning . . . 87
6.3 Simulation . . . 89
6.3.1 Evaluation on Dynamics Model Uncertainties . . . 90
6.3.2 Results of Transfer Learning on Empirical Model . . . 93

7 Conclusion 99
7.1 Summary of Contributions . . . 100
7.1.1 Learning Optimal Control under Excessive Time-Correlated Disturbances . . . 100
7.1.2 Transferring Disturbance Rejection Policy under Dynamics Model Mismatch . . . 101
7.2 Discussion and Future Work . . . 102
7.2.1 Dimensionality of Identification Model . . . 102
7.2.2 Theoretical Proof of Convergence . . . 102
7.2.3 Representation of Wave Disturbances . . . 103
7.2.4 Real Robot Deployment . . . 103
7.2.5 Online Model Learning . . . 104

Appendices 105

Bibliography 105

List of Figures

1.1 Examples of underwater robotic systems. 2

1.2 The flow of ideas in this thesis. 10

3.1 Working flow of joint learning algorithms for disturbance rejection control ($\overline{\boldsymbol{u}}$ and $\underline{\boldsymbol{u}}$ are the robot control constraints). 26

3.2 Submerged Pile Inspection Robot (SPIR) is designed at Centre for Autonomous Systems (CAS) of University of Technology Sydney (UTS), to operate in shallow bathymetry to clean infrastructure. 27

3.3 Simulated inverted pendulum. 31

3.4 Network architecture of HWRL. 33

3.5 Architecture of Disturbance Observer (DOB). 36

3.6 Architecture of Gated Recurrent Unit (GRU). 37

3.7 Network architecture of DOB-Net. 38

3.8 Simulated underwater robot (solid: current pose, transparent: target pose). 40

3.9 Example simulated wave disturbances (Type III) in X, Y, Z and yaw directions. 42

3.10 Example real wave disturbances in X, Y, Z and yaw directions. 42

3.11 Training reward: comparison of different disturbance strengths for Deep Deterministic Policy Gradient (DDPG). 43

3.12 Training reward: comparison of using history or not for HWRL. 44

3.13 Training reward: comparison of different history lengths for HWRL. 45

3.14 Comparison of 3D trajectories: (a) Robust Integral of Sign Error (RISE) controller; (b) DDPG policy; (c) HWRL policy. 46

3.15 Training rewards: (a) Type II simulated disturbances; (b) Type III simulated disturbances. 48

3.16 Distribution of the distance between the robot and the target with Type II simulated disturbances, the algorithms for comparison are given in Table 3.2. 49

3.17 Distribution of the distance between the robot and the target with Type III simulated disturbances, the algorithms for comparison are given in Table 3.2. 50

3.18 3D trajectories of the underwater robot with Type III simulated disturbances (R is the radius of the converged region). Note the trajectory optimization assumes that the disturbance dynamics is known in advance, thus provides the ideal performance. 51

3.19 Distribution of the distance between the robot and the target with real disturbances, the algorithms for comparison are given in Table 3.2. 52

3.20 3D trajectories of the underwater with real disturbances (R is the radius of the converged region). 53

4.1 Diagram of GCP-ODI. The online disturbance identification model (ODI) employs Recurrent Neural Network (RNN) to identify the disturbance parameters $\boldsymbol{\mu} = (A_1, \omega_1, \phi_1, \cdots, A_k, \omega_k, \phi_k)$ from the past states and actions, and the disturbance prediction model (DP) formulates a sequence of future disturbances $\boldsymbol{d}_{t:t+n}$ from these parameters, where $\boldsymbol{d}_t = A_1 \sin(\omega_1 t + \phi_1) + \cdots + A_k \sin(\omega_k t + \phi_k)$. The generalized control policy (GCP) then takes the predicted future disturbances $\boldsymbol{d}_{t:t+n}$ along with the current state $\boldsymbol{x}_t$ to compute the control action $\boldsymbol{u}_t$. 56

4.2 Example simulated wave disturbances (Type III) in X, Y and yaw directions. 61

4.3 Distribution of the distance between the robot and the target during last 100 timesteps for GCP-ODI: (a) Advantage Actor Critic (A2C); (b) GCP-true using as input the frequency domain signals $(A_1, \omega_1, \phi_1, \cdots, A_k, \omega_k, \phi_k)$; (c) GCP-true using as input the processed frequency domain signals $(A_1, \omega_1, \omega_1 t + \phi_1, \cdots, A_k, \omega_k, \omega_k t + \phi_k)$; (d) GCP-true using as input the time domain signals $\boldsymbol{d}_{t:t+n}$; (e) GCP-ODI at the 1st iteration; (f) GCP-ODI at the 2nd iteration; (g) GCP-ODI at the 3rd iteration; (h) GCP-ODI at the 4th iteration; (i) DOB-Net. 62

4.4 Trajectories of the robot using A2C and GCP-ODI with simulated dynamics model. 63

5.1 Working flow of model-based initialization of model-free reinforcement learning (HPA-init). 70

5.2 Compensation of model-based policy with model-free policy (HPA-comp). . 71

5.3 Diagram of the visual localization system. 73

5.4 Software interface of the visual localization system. 74

5.5 Field test of the SPIR over the QR code markers. 75

5.6 Thruster layout of the SPIR. 76

5.7 Thruster dynamics (mapping between Pulse Width Modulation (PWM) and thrust force) of Blue Robotics T200 Thruster. 77

5.8 Distributions of the distance between the robot and the target during last 100 timesteps: (a) comparison among iLQR, model-based policy and model-free policy, all run in the source task with the simulated disturbances; (b) comparison among purely model-free policy, HPA-comp policy and HPA-init policy in the target task. 79

5.9 Training rewards of the purely model-free RL and HPA: (a) training in the target task without disturbances, both model-based policy and model-free policy are constructed using feedforward network, the model-based policy are only used as the initialization for the model-free adaptation; (b) training in the target task with disturbances, two different algorithms for HPA are compared, either using the simulated disturbances in the source task or not. 81

5.10 State-action distributions in X-axis of the model-based policy and HPA policy: (a) learning model-free compensatory policy; (b) model-based initialization of model-free RL. 82

6.1 Diagram of compensatory control learning (CCL). 86
6.2 Diagram of transition mismatch learning with control combination (TML-control). 87
6.3 Diagram of transition mismatch learning with feature combination (TML-feature). 88
6.4 Distribution of the distance between the robot and the target during last 100 timesteps for different model variations: (1) GCP-ODI trained and tested on the original model (i.e., source policy); (2) GCP-ODI trained and tested on the model variation (i.e., target policy); (3) GCP-ODI trained on the original model and tested on the model variation (i.e., unadapted policy); (4) transfer learning using TML-feature (i.e., adapted policy). . 91
6.5 Distribution of the distance between the robot and the target during last 100 timesteps for the transfer learning: (a) GCP-ODI trained and tested on the mathematical model; (b) GCP-ODI trained and tested on the empirical model; (c) GCP-ODI trained on the mathematical model and tested on the empirical model; (d) transfer learning using CCL; (e) transfer learning using TML-control; (f) transfer learning using TML-feature; (g) transfer learning using HPA-init. 93
6.6 Training process of the transfer learning among different algorithms: (a) cumulative task reward; (b) cumulative transition mismatch. 94
6.7 Comparison among different control signals (X-axis) for the robot during the transfer learning on the empirical model: (a) transfer learning using TML-control; (b) transfer learning using TML-feature. 95
6.8 Trajectories of the underwater robot using GCP-ODI and TML-feature under the empirical model. 96
6.9 Comparison between the real disturbances and the predicted disturbances by ODI in Y direction, during transfer learning on the empirical model. . . 97

List of Tables

3.1 Distributions of disturbance parameters used in simulation. 41
3.2 Different algorithms for disturbance rejection control. 47

6.1 Distributions of disturbance parameters used in the source task. 89
6.2 Variations in the mathematical model. 90
6.3 Predicted disturbance parameters by ODI. 97

Acronyms & Abbreviations

UTS	University of Technology Sydney
CAS	Centre for Autonomous Systems
AUV	Autonomous Underwater Vehicle
ROV	Remotely Operated Vehicle
SPIR	Submerged Pile Inspection Robot
IMU	Inertial Measurement Unit
DOF	Degree of Freedom
TCM	Thruster Control Matrix
PWM	Pulse Width Modulation
DOB	Disturbance Observer
DOBC	Disturbance Observer Based Control
RISE	Robust Integral of Sign Error
MPC	Model Predictive Control
LQR	Linear Quadratic Regulator
iLQR	iterative Linear Quadratic Regulator
EKF	Extended Kalman Filter
MDP	Markov Decision Process

POMDP	Partially Observable Markov Decision Process
TD	Temporal Difference
DQN	Deep Q-Network
DDPG	Deep Deterministic Policy Gradient
A2C	Advantage Actor Critic
TRPO	Trust Region Policy Optimization
PPO	Proximal Policy Optimization
SGD	Stochastic Gradient Descent
RNN	Recurrent Neural Network
GRU	Gated Recurrent Unit