ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

DEEP REINFORCEMENT LEARNING FOR PARTIALLY OBSERVABLE MARKOV DECISION PROCESSES

Ph.D. THESIS

Mehmet HAKLIDIR

Department of Control and Automation Engineering

Control and Automation Engineering Programme

JULY 2022

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

DEEP REINFORCEMENT LEARNING FOR PARTIALLY OBSERVABLE MARKOV DECISION PROCESSES

Ph.D. THESIS

Mehmet HAKLIDIR
(504102110)

Department of Control and Automation Engineering

Control and Automation Engineering Programme

Thesis Advisor: Prof. Dr. Hakan TEMELTAŞ

JULY 2022

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

KISMİ GÖZLEMLENEBİLİR MARKOV KARAR SÜREÇLERİ İÇİN DERİN PEKİŞTİRMELİ ÖĞRENME

DOKTORA TEZİ

Mehmet HAKLIDIR
(504102110)

Kontrol ve Otomasyon Mühendisliği Anabilim Dalı

Kontrol ve Otomasyon Mühendisliği Programı

Tez Danışmanı: Prof. Dr. Hakan TEMELTAŞ

TEMMUZ 2022

Mehmet HAKLIDIR, a Ph.D. student of ITU Graduate School student ID 504102110, successfully defended the dissertation entitled “DEEP REINFORCEMENT LEARNING FOR PARTIALLY OBSERVABLE MARKOV DECISION PROCESSES”, which he prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :	**Prof. Dr. Hakan TEMELTAŞ** İstanbul Technical University	....
Jury Members :	**Prof. Dr. Osman Nuri UÇAN** Altınbaş University	....
	Assoc. Prof. Dr. Gülay ÖKE GÜNEL Istanbul Technical University	....
	Assoc. Prof. Dr. Volkan SEZER Istanbul Technical University	....
	Prof. Dr. Ahmet YAZICI Eskisehir Osmangazi University	....

Date of Submission : 21 June 2022
Date of Defense : 19 July 2022

To my better half,

FOREWORD

I would like to thank my supervisor Prof. Dr. Hakan TEMELTAŞ for invaluable advice and support during the thesis study. I would also like to thank to my thesis progress members Prof. Dr. Osman Nuri UÇAN, Assoc. Prof. Dr. Gülay Öke GÜNEL and jury members Assoc.Prof. Dr. Volkan SEZER, Prof. Dr. Ahmet YAZICI for their valuable comments and guidance to my studies. I would also like to express my deepest gratitude to Prof. Dr. İbrahim EKSİN who provided moral support to continue at the critical point in the first phase of my doctorate study.

My sincere thanks to my dear wife Füsun Servin TUT HAKLIDIR and Nis HAKLIDIR, who have been with me with all their love and faith throughout my doctoral process.

July 2022 Mehmet HAKLIDIR
(M. Sc.)

TABLE OF CONTENTS

Page

ABBREVIATIONS

ADRQN : Action-specific Deep Recurrent Q-Network
ARM : Advantage-based Regret Minimization
App : Appendix
COPOS : Compatible Policy Search
DDPG : Deep Deterministic Policy Gradient
DDRQN : Deep Distributed Recurrent Q-Network
DNN : Deep Neural Network
DP : Direct Policy
DQN : Deep Q-Network
DRQN : Deep Recurrent Q-Network
DRL : Deep Reinforcement Learning
DVRL : Deep Variational Reinforcement Learning
ELBO : Evidence Lower Bound
GIDAS : German In-Depth Accident
GPS : Guided Policy Search
KL : Kullback-Leibler
LSTM : Long Short-Term Memory
MDP : Markov Decision Process
MOMDP : Mixed Observable Markov Decision Processes
MPC : Model Predictive Control
NavA3C : Navigation Asynchronous Advantage Actor Critic
NN : Neural Network
PILCO : Probabilistic Inference for Learning Control
POMDP : Partially Observable Markov Decision Process
PPO : Proximal Policy Optimization
RAM : Random Access Memory
RDPG : Recurrent Deterministic Policy Gradient
ReLU : Rectified Linear Units
RL : Reinforcement Learning
RNN : Recurrent Neural Networks

SAC : Soft Actor-Critic
SVG : Stochastic Value Gradients
TD3 : Twin Delayed DDPG
TRPO : Trust Region Policy Optimizati

SYMBOLS

$\boldsymbol{\alpha}$	: Entropy Temperature
$\mathbf{a}$	: Action
$\boldsymbol{\mathcal{A}}$	: Action Space
$\mathcal{D}$	: Replay Buffer
$\boldsymbol{h_t}$	: History
$\mathcal{H}$	: Entropy
$\mathbf{o}$	: Observation
$\boldsymbol{O}$	: Observation Space
$\boldsymbol{\mathcal{O}}$	: Observation Model
$\boldsymbol{r}$	: Reward
$\boldsymbol{R}$	: Reward Function
Q	: Q-Function
$\mathbf{s}$	: State
$\mathbf{s'}$	: Next State
$\boldsymbol{S}$	: State Space
$\mathbf{t}$	: Time
$\boldsymbol{T}$	: Transition Model
V	: Value Function
γ	: Discount Factor
$\mu_0(\boldsymbol{s})$	: Initial State Distribution
π	: Policy Function
ϵ_t	: Noise Vector
$\tilde{a}_\theta(s)$	: Policy Sample
υ	: Speed

LIST OF TABLES

Page

LIST OF FIGURES

Page

DEEP REINFORCEMENT LEARNING FOR PARTIALLY OBSERVABLE MARKOV DECISION PROCESSES

SUMMARY

Deep reinforcement learning has recently gained popularity owing to its many successful real-world applications in robotics and games. Conventional reinforcement learning faces a substantial challenge in developing effective algorithms for high-dimensional environments. The use of deep learning as a function approximator in reinforcement learning is a viable solution to overcome this challenge. Furthermore, in deep reinforcement learning, the environment is often thought to be fully observable, meaning that the agent can perceive the true state of the environment and so act appropriately in the current state. Most real-world problems are partially observable and the environmental models are unknown. Therefore, there is a significant need for reinforcement learning approaches to solve these problems, in which the agent perceives the state of the environment partially and noisily. Guided reinforcement learning methods solve this issue by providing additional state knowledge to reinforcement learning algorithms during the learning process, thereby allowing them to solve a partially observable Markov decision process (POMDP) more effectively. However, these guided approaches are relatively rare in the literature, and most existing approaches are model-based, which means that they require learning an appropriate model of the environment first.

In this thesis, we present a novel model-free approach that combines the soft actor-critic method and supervised learning concept to solve real-world problems, formulating them as POMDPs. We evaluated our approach using modified partially observable MuJoCo tasks. In experiments performed on OpenAI Gym, an open-source simulation platform, our guided soft actor-critic approach outperformed other baseline algorithms, gaining 7~20% more maximum average return on five partially observable tasks constructed based on continuous control problems and simulated in MuJoCo.

To solve the autonomous driving problem, we focused on decision making under uncertainty, as a partially observable Markov decision process, using our guided soft actor-critic approach. A self-driving car was trained in a simulation environment, created using MATLAB/SIMULINK, for a scenario in which it encountered a pedestrian crossing the road. Experiments demonstrate that the agent exhibits desirable control behavior and performs close to the fully observable state under various uncertainty situations.

KISMİ GÖZLEMLENEBİLİR MARKOV KARAR SÜREÇLERİ İÇİN DERİN PEKİŞTİRMELİ ÖĞRENME

ÖZET

Derin pekiştirmeli öğrenme, robotik ve oyunlardaki birçok başarılı gerçek dünya uygulaması sayesinde son zamanlarda popülerlik kazanmıştır. Geleneksel pekiştirmeli öğrenme, yüksek boyutlu ortamlar için etkili algoritmalar geliştirmede önemli bir zorlukla karşı karşıyadır. Derin öğrenmenin pekiştirmeli öğrenmede bir fonksiyon tahmincisi olarak kullanılması, bu zorluğun üstesinden gelmek için iyi bir yaklaşımdır. Derin pekiştirmeli öğrenmede, çevrenin genellikle tamamen gözlemlenebilir olduğu düşünülür ki, bu da etmenin çevrenin gerçek durumunu algılayabileceği ve dolayısıyla mevcut duruma uygun şekilde hareket edebileceği anlamına gelir. Gerçek dünya problemlerinin çoğu esasen kısmen gözlemlenebilir olup ve çevresel model bilinmemektedir. Bu nedenle, etmenin çevrenin durumunu kısmen ve gürültülü olarak algılaması nedeniyle bunları çözmek için pekiştirmeli öğrenme yaklaşımlarına önemli bir ihtiyaç vardır. Literatürde, çoğu derin pekiştirmeli öğrenme yaklaşımı Markov karar süreçlerine uygulanmıştır. Ancak kısmi gözlemlenebilir Markov karar süreçleri için derin pekiştirmeli öğrenme yeterince çalışılmamıştır. Rehberli pekiştirmeli öğrenme yöntemleri, öğrenme sürecinde pekiştirmeli öğrenme algoritmalarına ek durum bilgisi sağlayarak bu sorunu çözer ve kısmen gözlemlenebilir Markov karar sürecini (POMDP) daha etkin bir şekilde çözmelerine olanak tanır. Bununla birlikte, bu rehberli yaklaşımlar literatürde nispeten nadirdir ve mevcut yaklaşımların çoğu model tabanlıdır, yani öncelikle uygun bir çevre modelinin öğrenilmesini gerektirir.

Rehberli politika arama yaklaşımları, önceki kısaltılmış gözlem-eylem çiftlerini mevcut durum-eylem temsillerine dahil ederek kısmi gözlemlenebilir problemleri çözmek için pekiştirmeli öğrenme ve gözetimli öğrenmeyi birleştirmek için ilham vericidir. Model tabanlı yaklaşımların, gerçek dünyada gereken etkileşim miktarını azaltmak için dinamik bir çevre modelini öğrenmesi gerekir. Ancak, bu karmaşık görevler için zorlayıcı olabilir ve kapsamlı hesaplama gerektirir. Öte yandan, modelsiz yaklaşımlar, bir politikayı doğrudan çevre ile etkileşimlerden öğrenir ve çevrenin tam bir temsiline ihtiyaç duymadıkları için çok daha fazla esnekliğe sahiptir. Modelsiz yaklaşımları kullanarak etkili sonuçlara ulaşmak için, görevin karmaşıklığına bağlı olarak çevre ile yeterli sayıda etkileşim sağlanmalıdır. Bu nedenle, çoğu geleneksel rehberli politika arama yönteminin aksine, bu çalışamada modelsiz bir yaklaşıma odaklanılmıştır.

Bu tez çalışmasında, gerçek dünya sorunlarını çözmek için soft aktör-kritik yöntemini ve denetimli öğrenme kavramını birleştiren ve bunları POMDP'ler olarak formüle eden yeni bir modelsiz yaklaşım önerilmektedir.

Yaklaşımımızda, soft aktör-kritik'in orijinal konseptinden farklı olarak, eğitim aşamasında, rehberli soft aktör-kritik mimarisi iki aktör ve bir eleştirmenden oluşur. Örnekleri oluşturmak için iki aktör aynı ortamla yinelemeli olarak etkileşime girer. Son kontrol aktörü, eylemleri kısmi gözlemlere dayalı olarak seçerken, rehberlik eden aktör eylemleri tam durum gözlemlerini algılayarak seçmektedir. Hem nihai kontrol hem de rehberlik politikaları hem beklenen ödülü hem de entropiyi en üst düzeye çıkarmak için eğitilmiştir. İlaveten, bir derin yapay sinir ağı, kontrol politikası aktörüne yol gösterici bir politika aktörü olarak hareket etmeyi öğretmek için eğitilmiştir.

Yaklaşımımızda, çeşitli politikaların ürettiği örnekleri kullanmak için iki politikanın dağılımlarının benzer olması gerekmektedir. Bu nedenle, yöntemin performansını ve kararlılığını geliştirdiği ve aynı ilkeye yakınlaşmalarını garanti ettiği için ayarlanabilir entropi sıcaklığı α'yı kullanmak için ek bir güncelleme adımı eklenmiştir.

Yaklaşımımızın performansını, değiştirilmiş, kısmen gözlemlenebilir MuJoCo görevlerini kullanarak değerlendirdik. Açık kaynaklı bir simülasyon platformu olan OpenAI Gym'de gerçekleştirilen deneylerde, rehberli soft aktör-kritik yaklaşımımız diğer temel algoritmalardan daha iyi performans göstermiş ve sürekli kontrol problemlerine dayalı olarak oluşturulan ve MuJoCo'da simüle edilen kısmen gözlemlenebilir beş görevde %7~20 daha fazla maksimum ortalama getiri elde etmiştir. Önerilen rehberli Soft Aktör Kritik yaklaşımımız, klasik Soft Aktör Kritik için önerilen yeni mimarimiz ve etkileşim döngüsü sayesinde kısmen gözlemlenebilir sürekli kontrol deneylerinde karşılaştırma yöntemlerinden daha iyi performans göstermektedir.

Bu tez çalışması kapsamında geliştirilen rehberli soft aktör-kritik yaklaşımımızın performansının gözlemlenebilmesi için, sürekli zamanlı robotik görevleri dışında, belirsizlik altında otonom sürüş problemine uygulanması için deneyler de gerçekleştirilmiştir.

Belirsizlik altında otonom sürüş, gerçek dünyada durumların ve eylemlerin yüksek boyutlu uzayda olduğu zorlu bir problemdir. Bu görevleri için, otonom aracın herhangi bir statik engel veya diğer dinamik trafik kullanıcıları (araçlar ve yayalar) ile çarpışmadan hedefe ulaşması beklenmektedir. Yoldaki yayaların veya diğer araçların davranışları ve hava koşulları yüksek belirsizlik içermektedir. Yayaların davranışlarını tahmin etmek zordur ve diğer trafik unsurlarının davranışlarının belirsizliği de kazalara neden olabilir. Dinamik nesnelerin davranışı zamanla değişen ve tahmin edilemezken, otonom aracın kısmi gözlemlerle eylemlere karar vermesi gerekir ki, bu da problemin zorluk derecesini artırmaktadır. Bu nedenle, belirsizlik altında mevcut duruma en uygun kararların verilebilmesi için daha uzun bir geçmiş verisini değerlendirmek gereklidir. Aksi takdirde, araç "kör" alana girdiğinde yayaların bilgilerini tamamen kaybedebilir.

Otonom araçların belirsizlik altında karar verme problemi, otonom araç tarafından gerçek zamanlı olarak çevrimiçi çözülecek, kısmen gözlemlenebilir Markov karar süreci (POMDP) olarak modellenebilir.

Bu çalışmada, otonom aracın kısmen veya tamamen çevreyi gözlemleme yeteneğini kaybettiği durumlara odaklanılmıştır. Çalışmada, geliştirdiğimiz rehberli soft aktör-kritik yöntemimiz kullanılarak, belirsizlik altında karar verme odaklı otonom sürüş problemini kısmi gözlemlenebilir Markov karar süreci olarak çözmek için yeni bir yaklaşım önerilmektedir. Deneyler, MATLAB/SIMULINK üzerinde gerçekleştirilmiş ve otonom aracın ve yayanın başlangıç hızlarının, birbirlerine olan başlangıç mesafenin farklı olduğu, otonom aracın yoldan karşıya geçen bir yaya ile karşılaştığı senaryolar oluşturularak eğitim ve testler gerçekleştirilmiştir. Rehberli soft aktör kritik çözümünün, kısmi gözlemlenebilir durumlar karşısındaki performansını tam gözlemlenebilir durumlar ile karşılaştırılması için 1000 farklı koşum ile performans testler gerçekleştirilmiş ve sonuçları sunulmuştur. Test sonuçları, kontrol etmeninin istenen kontrol davranışını sergilediğini ve çeşitli belirsiz durumları altında tamamen gözlemlenebilir duruma yakın bir şekilde gerçekleştirildiğini göstermektedir.

1. INTRODUCTION

Recently, deep learning techniques have been implemented in reinforcement learning (RL) as function approximators, yielding deep RL, to solve high-dimensional, continuous state-space problems (Azizzadenesheli et al., 2015), (Lillicrap et al., 2015), (Mnih et al., 2015), (Schulman et al., 2015). In Markov decision process (MDP)-based deep RL approaches, the agent is generally assumed to be capable of fully observing the true state of the environment and that the environment's next state is solely determined by the current state and the action taken. This ensures that the agent can act in the best manner by simply reacting to the current state of the environment. But, in many real-world problems, this assumption is incorrect. Because, the agent often perceives the state of the real-world environment partially and noisily. Another way of saying that the Markov property is seldom found in real-world environments. A reactive policy found by the MDP cannot be said to be optimal if the observations do not hold the Markov property. In such cases, the partially observable Markov decision process (POMDP) approach is used. POMDP is an MDP extension by allowing an agent to represent a subsequent decision-making process under uncertainty.

The development of RL algorithms for solving POMDP poses several challenges. The agent explicitly observes the state transitions in the MDP, and the generative model is easily estimated using empirical estimators. In POMDP, partial and noisy observations must be used for the transition and reward models. POMDPs require a range of observation probabilities outside of direct observation and a probability distribution across all possible situations based on MDP to predict the next state. The optimal policy for POMDP can be calculated by solving an augmented MDP based on a continuous belief domain. Finally, it is not trivial to integrate prediction and planning with guarantees in a discovery, and no regret strategy is currently available. Guided RL methods solve this issue by providing additional state knowledge to RL algorithms during the learning process, thereby allowing them to solve POMDPs more effectively.

In the literature, most deep RL approaches have been applied to Markov decision processes. Deep RL for POMDPs has not been sufficiently studied. Following the successful performance of the DQN (Deep Q-network), several approaches have been developed for POMDPs, mainly focusing on remembering important observations and implementing RNNs (Recurrent neural networks) (Egorov, 2015), (Foerster et al., 2016), (Hausknecht and Stone, 2015), (Zhu et al., 2017). However, especially in real-world scenarios, there are also partially observable environments in which the agent must consider account information gathering and therefore engage in much more exploration and interaction with these environments to learn how to collect and use information. Few studies have been conducted on these scenarios (Heess et al., 2015), (Wang et al., 2017), (Azizzadenesheli et al., 2018), (Igl et al., 2018), (Jin et al., 2018), (Le et al., 2018), (Song et al., 2018), (Pajarinen et al., 2019).

1.1 Purpose of Thesis

In this study, we present a novel approach that combines the soft actor-critic (SAC) method (Haarnoja et al., 2018) and supervised learning concepts to solve real-world problems, and formulate them as POMDPs. In experiments performed on OpenAI (2021), an open-source simulation platform, our guided SAC approach outperformed other baseline algorithms trained directly on five partially observable tasks constructed based on continuous control tasks in the MuJoCo (Todorov et al., 2012). The main contributions of this study are as follows: Our proposed guided SAC approach outperforms the comparison methods in partially observable continuous control experiments owing to our proposed novel architecture and interaction loop for the classic soft actor–critic. Unlike the original concept of the SAC, in the training phase, the guided SAC architecture consists of two actors and a critic. To generate the samples, the two actors interact iteratively with the same environment. The final control actor chooses actions based on partial observations, whereas the guiding actor chooses actions by perceiving the full-state observations. Both the final control and guiding policies are trained to maximize both the expected reward and entropy. Furthermore, a deep neural network (DNN) is trained to teach the control policy actor to act as a guiding policy actor.

1.2 Literature Review

In this section, we first summarize the taxonomy of deep reinforcement learning (DRL) algorithms. Second, we provide a detailed literature survey of deep reinforcement learning approaches for POMDPs.

1.2.1 Taxonomy of deep reinforcement learning algorithms

Reinforcement learning is a machine learning subfield in which an agent attempts to learn actions by maximizing the reward received through interaction with the environment (Sutton and Barto, 2018). Conventional RL policies such as tabular, linear, and radial basis networks have a number of drawbacks, especially in terms of high-level behavior representation and the computational cost of updating parameters. Combining RL and deep learning can extract more information from raw inputs by passing it through multiple neural layers, which can have a large number of parameters and represent highly nonlinear problems (Heess et al., 2015). Using deep learning as a function approximator in RL (Aggarwal, 2018), deep RL methods can approximate optimal policies and/or value functions.

The following are the several types of deep reinforcement learning approaches: model-free/model-based, value-based/policy-based, and on-policy/off-policy. Figure 1.1 shows the taxonomy of the deep reinforcement learning methods.

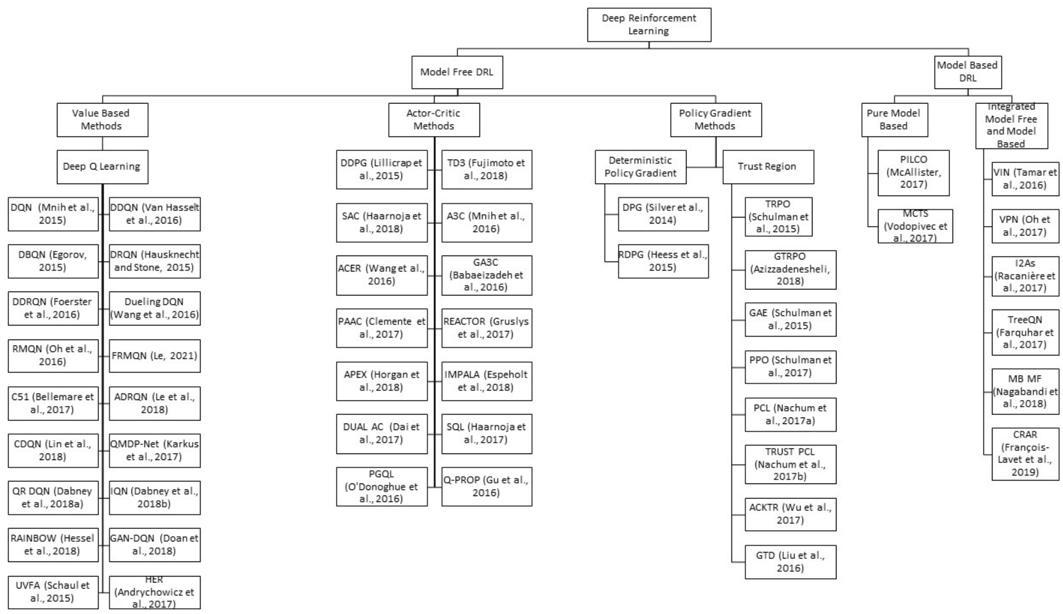


Figure 1.1 : Taxonomy of Deep Reinforcement Learning Algorithms.

In real-world problems, interaction with the physical world may be restricted for a variety of reasons such as safety and cost. The amount of interaction required in the real world can be reduced by learning the dynamical model of the environment. Furthermore, exploration can be performed on the learned models. With model-based approaches, the agent tries to learn the transition and reward functions that are utilized when taking an action. Gathering information on the dynamics of the environment is required to develop a model approach. Unlike model-based approaches, model-free approaches do not require this type of information. Model-free approaches directly illustrate the underlying MDP to learn an unknown model. Model-free policy search methods require more training data and longer training time than model-based methods (Kiran et al, 2021).

Value-based methods estimate value functions directly and infer the policies that should be implemented. These value functions are calculated by computing the immediate reward and discounting the next state value with the previous state. Policy-based methods specifically discover policies that can take action without value function values (Kiran et al, 2021). The distinction between value- and policy-based methods is based on the trade-off between better performance and more complex strategies. Both methods need to propose actions and evaluate their outcomes. However, value-based methods emphasize determining the maximum cumulative return and have a policy based on recommendations, which try to directly establish the optimal policy that maximizes the cumulative discounted reward. Furthermore, actor-critic approaches are also known as hybrid approaches that combine value and policy-based methods (Kiran et al, 2021).

On-policy algorithms train agents based exclusively on the current policy experience. They are sample-inefficient in the context of changing an agent's policy and actions through training. This ensures that prior experience cannot be considered for training purposes. Off-policy algorithms can employ the experience gained from other policies (Kardell and Kuosku, 2017).

1.2.2 Deep reinforcement learning for partially observable Markov decision process

In the literature, most deep reinforcement learning (DRL) approaches have been applied to Markov decision processes (MDP). The DRL for POMDPs has not been sufficiently studied. POMDPs for discrete have been studied using deep RL approaches (Egorov, 2015), (Hausknecht and Stone, 2015), (Foerster et al., 2016), (Zhu et al., 2017), (Jin et al., 2018), (Le et al., 2018) and continuous (Heess et al., 2015), (Wang et al., 2017), (Azizzadenesheli et al., 2018), (Igl et al., 2018), (Jin et al., 2018), (Le et al., 2018), (Song et al, 2018), (Pajarinen et al, 2019), (Azizzadenesheli et al., 2019) control tasks. (Table 1.1).

Table 1.1 : Literature Survey for Deep Reinforcement Learning Methods for POMDPs.

Authors	Method	Observability	Value /Policy Based	On/Off Policy	Action Space
(Egorov, 2015)	DBQN	model-based POMDP	Value Based	Off Policy	Discrete
(Hausknecht and Stone, 2015)	DRQN	model-free POMDP	Value Based	Off Policy	Discrete
(Foerster et al., 2016)	DDRQN	model-free POMDP	Value Based	Off Policy	Discrete
(Zhu et al., 2018)	ADRQN	model-free POMDP	Value Based	Off Policy	Discrete
(Jin et al., 2018)	ARM	model-free POMDP	Value Based	Off Policy	Discrete
(Le et al., 2018)	hDRQN	model-free POMDP	Value Based	Off Policy	Discrete
(Heess et al., 2015)	RDPG	model-free POMDP	Policy Based	Off Policy	Continuous
(Wang et al., 2017)	Fast-RDPG	model-free POMDP	Policy Based	Off Policy	Continuous
(Song et al., 2018)	RDPG	model-free POMDP	Policy Based	Off Policy	Continuous
(Igl et al., 2018)	DVRL	model-free POMDP	Policy Based	Off Policy	Continuous/Discrete
(Pajarinen et al., 2019)	COPOS	model-free POMDP	Policy Based	On Policy	Continuous/Discrete
(Azizzadenesheli et al., 2019)	GTRPO	model-free POMDP	Policy Based	On Policy	Continuous

Egorov (2015) suggested an extension of the DQN approach and mapped an action-observation history to an optimal action to solve POMDPs. Egorov (2015) showed that DQNs can learn good policies but require substantially more computational resources, and while the Q values converge, the policies are vulnerable to minor disturbances. Another approach, DRQN (Deep Recurrent Q-network), can be implemented in a model-free representation using additional LSTM layers (Hausknecht and Stone, 2015). A qualified policy network can capture all historical data in the LSTM layer, and the output actions are based on all historical observations. After performing extensive analysis on DRQN, Foerster et al. proposed DDRQN (Deep Distributed Recurrent Q-networks), which used both prior observations and actions as inputs. The action-specific deep recurrent Q-network (ADRQN) uses action-observation pairs to obtain an optimal policy in partially observable environments (Zhu et al., 2018). Jin et al. proposed a novel algorithm called ARM, which does not require access to a Markovian state and is based on the notion of regret minimization. They demonstrated that regret estimation approaches are better at handling partial observability than regular RL. Le et al. proposed a hierarchical deep RL approach for learning in a hierarchical POMDP. Instead of using DDPG (Deep Deterministic Policy Gradient) for continuous regulation, Heess et al. proposed RDPG (Recurrent Deterministic Policy Gradient) for the POMDP, which involves using recurrent neural networks (RNNs) to approximate the actor and critic. Wang et al. (2017) modeled the autonomous navigation of an unmanned aerial vehicle (UAV) under uncertainty in a large-scale complex environment as a POMDP. Wang et al. proposed a fast-RDPG using an actor-critic-based method with function approximation. (Song et al., 2018) extended RDPG to solve partially observable bipedal locomotion tasks; however, only one was investigated. The fast-RDPG directly maximizes the projected long-term accumulated discounted reward to generate policy updates. Igl et al. proposed deep variational RL (DVRL) using inductive biases in a network to develop a generative model capable of mimicking belief updates. DVRL uses the evidence lower bound (ELBO) loss to estimate belief states directly for recurrent RL policies, whereas probabilistic inference for learning control (PILCO) uses dynamic models learned from beliefs rather than observations. Previous studies have focused on partial observations, but none has discussed stale observations. Pajarinen et al. proposed Compatible Policy Search (COPOS), a variant of trust region policy optimization (TRPO), which has a transformer network using Adam optimization and applies to machine translation tasks. Although the original TRPO and proximal policy optimization (PPO) were planned for an MDP with an infinite horizon, they have recently been expanded for a finite horizon (Azizzadenesheli et al., 2019).

In this study, we propose a novel architecture and interaction loop for SAC (Haarnoja et al., 2018), which is a state-of-the-art methods for MDP, inspired by of (Zhang et al., 2016b) to solve partially observable continuous control tasks more efficiently in much more exploration and interaction with the environment by using a guided policy search approach to optimize the policy.

1.3 Contributions

The main contributions of this study are as follows:

- We propose a novel guided reinforcement learning method by providing additional state knowledge to reinforcement learning algorithms during the learning process, allowing them to solve a POMDP more effectively.

- Our proposed guided SAC approach outperforms the comparison methods in partially observable continuous control experiments owing to our proposed novel architecture and interaction loop for the classic soft actor–critic.
- On decision making under uncertainty for autonomous driving, our proposed guided SAC approach exhibited the desired control behavior and had a near-full observable performance in various uncertainty situations.

In our approach, to use the samples produced by various policies, the distributions of the two policies should be similar. We further introduced an additional update step to use the adjustable entropy temperature α, as it improves the performance and stability of the method and guarantees that they converge to the same policy.

1.4 Thesis Outline

This thesis is organized into seven chapters, including an introduction chapter. In the second chapter, the general formulation of sequential decision problems in stochastic environments under uncertainty is discussed. This chapter assists in comprehending the task's complexity as well as the motivations, ideas, and methods underlying the novel approach developed in the fourth chapter. In the third chapter, we introduce how to use deep reinforcement learning algorithms to solve continuous control problems with stochastic policies and how to improve guided RL methods on stochastic policies to solve POMDPs more effectively. The fourt chapter provides a detailed explanation of the proposed methodology. The fifth chapter provides the results obtained and discusses the results for experiments on partially observable robotic tasks constructed based on continuous control problems. The sixth chapter provides the results obtained and discusses the results for experiments on decision making under uncertainty for autonomous driving. The seventh chapter presents the conclusion and future work plan of this study.

2. SEQUENTIAL DECISION MAKING UNDER PARTIALLY OBSERVABLE ENVIRONMENT

2.1 Introduction

Many critical real-world problems necessitate a sequence of decisions made by the decision maker. As in single decision making, the decision maker should choose the action that maximizes the expected utility (Morgenstern and Von Neumann, 1953), but reasoning about future actions and observations is required for optimal decision making.

The decision maker, called the agent, take actions depending on observations of the environment. An observation-action cycle governs the agent's interaction with the environment, as depicted in Figure 2.1.

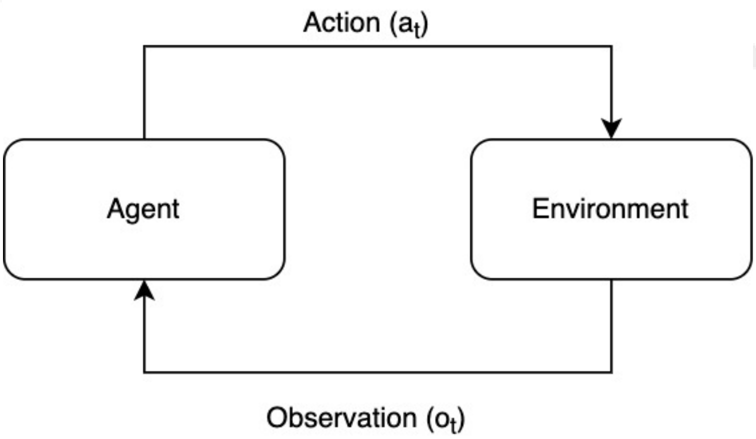


Figure 1.2 : The agent's interaction with the environment.

The agent receives an observation of the environment at time t, denoted by o_t. In most real-world problems, the observations are often partial or noisy. The agent then uses a decision-making process to determine an action. This may have a stochastic effect on the environment. The agent should choose an action that best fulfills its goals based on the previous sequence of observations and knowledge of the environment. (Kochenderfer, 2015).

In this chapter, the general formulation of sequential decision problems in stochastic environments under uncertainty is discussed. This chapter assists in comprehending the task's complexity as well as the motivations, ideas, and methods underlying the novel approach developed in Chapter 4. In the next section, we provide preliminaries and establish the notations used in this study, and formally introduce our problem statement.

2.2 Sequential Decision Making in Stochastic Environments

Sequential decision making refers to a circumstance in which a decision maker observes a process periodically before making a final decision. In sequential decision-making, the aim is to find a rule called an optimal policy, that optimizes the decision to minimize losses or maximize benefits, considering observation costs. Optimal decision-making necessitates the consideration of future action and observation sequences (Diederich, 2001).

Sequential decision-making is applied whenever a dynamic system controlled by a decision maker makes sequential decisions. In this system, the effects of actions on the environment may be uncertain. In this case, MDPs (Markov Decision Processes) can represent sequential decision problems.

2.2.1 Markov decision process (MDP)

Reinforcement learning is a subfield of machine learning in which an agent tries to learn actions by maximizing the reward it receives from interacting with the environment (Sutton and Barto, 2018). A Markov decision process (MDP) is a stochastic sequential decision-making method used to solve reinforcement learning problems. MDP can determine which action the decision-maker should take based on the current state of the system and its environment. To choose the best next action, this decision-making process considers information from the environment, the agent's actions, and rewards. MDPs can be classified as either finite or infinite, and continuous or discrete, depending on the number of actions and states accessible and the frequency with which decisions are made. (Puterman, 1990).

A MDP is used to model sequential decision-making problems, in which an agent interacts with a stochastic environment (Puterman, 2014). It is defined as a 4-tuple (S,

$\mathcal{A}$, T, R) where S is the state space, $\mathcal{A}$ is the action space, T is the transition model, and R is the reward function, which consists of

- a set of states $\mathbf{s} \in S$,
- a set of actions $\mathbf{a} \in A$,
- a transition function defined as $T: S \times \mathcal{A} \times S \rightarrow [0,1]$, where the probability of ending up in the next state $\mathbf{s}'$ after action $\mathbf{a}$ in state s is denoted $T(\mathbf{s}, \mathbf{a}, \mathbf{s}')$,
- r, a reward function defined as $r: S \times \mathcal{A} \rightarrow \mathbb{R}$,

Figure 2.2 shows an MDP agent's interaction with the environment.

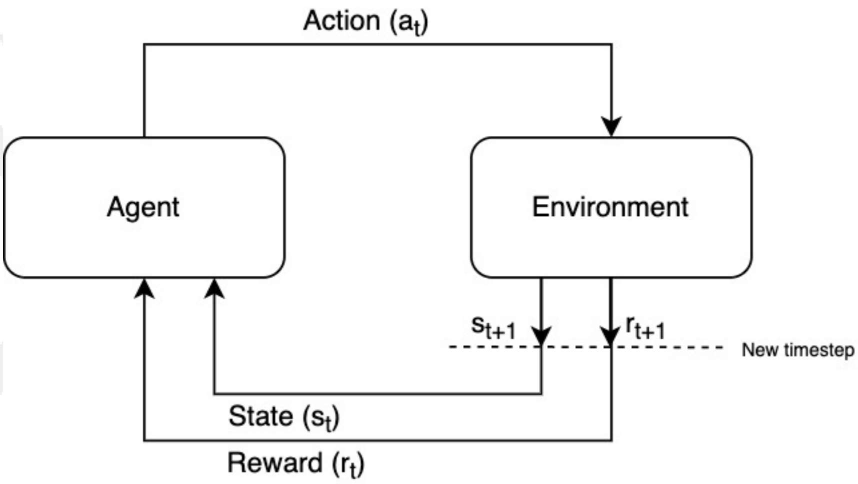


Figure 1.3 : An MDP agent's interaction with the environment.

2.2.2 Sequential decision making with state uncertainty

In the Markov decision process, the agent is generally assumed to be capable of fully observing the true state of the environment and that the environment's next state is solely determined by the current state and the action taken. This ensures that the agent can act in the best manner by simply reacting to the current state of the environment. But, in many real-world problems, this assumption is incorrect. Because, the agent often perceives the state of the real-world environment partially and noisily. Another way of saying that the Markov property is seldom found in real-world environments. A reactive policy found by the MDP cannot be said to be optimal if the observations do not hold the Markov property. In such cases, a partially observable Markov decision process (POMDP) can be used to model a sequential decision problem with state uncertainty. POMDP extends MDP by allowing an agent to represent a subsequent decision-making process under uncertainty.

2.2.2.1 Partially observable Markov decision process (POMDP)

In many real-world problems, the agent cannot fully observe the state of the current environment, which is necessary in MDP-based learning methods. The Markov property for MDPs does not hold in real-world scenarios in which observations are incomplete or noisy, such as autonomous driving, in which the agent can only observe parts of the environment and does not have a complete view of the world. Under uncertainty, the action results depend more on the current state signal. POMDPs generalize MDPs to decision-making processes under partial observability.

POMDP represents the dynamics of a wide range of real-world environments more effectively, because it obviously acknowledges that the agent's sensations are simply fragmentary hints at the system's basic state. (Hausknecht and Stone, 2015).

We assume that a POMDP is used to simulate the environment, and that the observations and actions taken from the beginning of the episode are adequate to determine the real state of the environment. A POMDP is formally defined (Spaan, 2012) as a tuple $(S, \mathcal{A}, O, T, \mathcal{O}, R, \gamma)$, where S is the state space, $\mathcal{A}$ is the action space, O is the observation space, T is the transition model, R is the reward function, $\mathcal{O}$ is the observation model, and γ is the discount factor, consisting of

- a set of states $\mathbf{s} \in S$,
- a set of actions $\mathbf{a} \in A$,
- a set of observations $\mathbf{o} \in O$ the agent can gain knowledge of its surroundings.,
- a transition function defined as $T: S \times \mathcal{A} \times S \rightarrow [0,1]$, where the probability of moving to the next state $\mathbf{s}'$ after action $\mathbf{a}$ in state s is denoted $T(\mathbf{s}, \mathbf{a}, \mathbf{s}')$,
- an observation function defined as $\mathcal{O}: S \times \mathcal{A} \times O \rightarrow [0,1]$, where $\mathcal{O}(\mathbf{s}', \mathbf{a}, \mathbf{o})$ is the probability of making observation $\mathbf{o}$ given that the agent took action $\mathbf{a}$ and landed in state $\mathbf{s}'$,
- r, a reward function defined as $r: S \times \mathcal{A} \rightarrow \mathbb{R}$,
- the initial state distribution $\mu_0(\boldsymbol{s})$ specifies the probability of state $\boldsymbol{s}$ being the initial state of the decision process.

- the history $h_t = (\mathbf{o}_0, \mathbf{a}_1, \ldots, \mathbf{a}_{t-1}, \mathbf{o}_t)$ as a vector containing all past observations and taken actions.

A POMDP is distinguished from a fully observable MDP by the agent's perception of an observation $\mathbf{o} \in O$ rather than the actual observation of the state of the environment. As a result, POMDPs have two more elements in their description than MDPs. The set of observations O represents all the possible sensors that the agent can receive. The observation function $\mathcal{O}$ indicates which particular observation the agent receives depending on the next state $\mathbf{s}'$ and perhaps also on its action $\mathbf{a}$ (Spaan, 2012).

Figure 2.3 shows a POMDP agent's interaction with the environment.

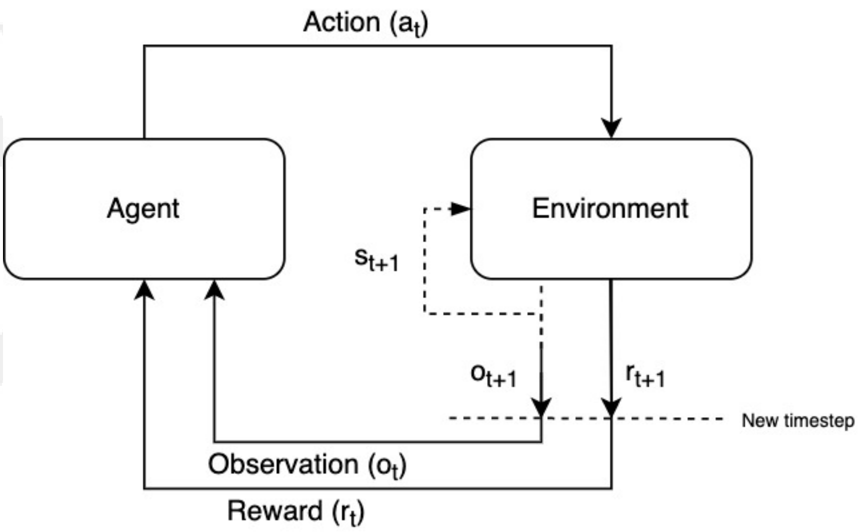


Figure 1.4 : A POMDP agent's interaction with the environment.

While the agent perceives the observation, the agent is unaware of the true status of the environment.

The agent interacts with the environment in discrete time-steps. In each time step t, it receives an observation $\mathbf{o} \in O$ based on the actual state $\mathbf{s} \in S$ of the environment and $\mathcal{O}$. Next, it performs an action $\mathbf{a} \in A$, receives a reward r according to R and proceeds to the next time step. When the agent reaches a terminal state, it begins a new episode in a new instance of the environment, starting at an arbitrary state.

The agent tries to maximize the expected return (equation 2.1) from each state s in time step t while also maximizing entropy (equation 2.2):

$$\max_{\pi \in \Pi} \mathbb{E}_{\rho_\pi}\left[\sum_{t=0}^{T} r(\mathbf{s}_t, \mathbf{a}_t)\right] \tag{2.1}$$

$$\max_{\pi \in \Pi} \mathbb{E}_{\rho_\pi}\left[\sum_{t=0}^{T} \alpha\mathcal{H}\big(\pi(\cdot|\, s_t)\big)\right] \tag{2.2}$$

We investigate environments with a multidimensional, continuous, bounded observation and action space as a possibility.

2.3 Chapter Conclusion

The agent explicitly observes the state transitions in the MDP, and the generative model is easily estimated using empirical estimators. In POMDP, partial and noisy observations must be used for the transition and reward models. POMDPs require a range of observation probabilities outside of direct observation and a probability distribution across all possible situations based on MDP to predict the next state. The optimal policy for POMDP can be calculated by solving an augmented MDP based on a continuous belief domain. Finally, it is not trivial to integrate prediction and planning with guarantees in a discovery, and no regret strategy is currently available. Guided RL methods solve this issue by providing additional state knowledge to RL algorithms during the learning process, thereby allowing them to solve POMDPs more effectively. In the next chapter, we explain how to use deep reinforcement learning algorithms for solving continuous control problems that are formulated as a POMDP with stochastic policies and how to improve guided RL methods on stochastic policies to solve POMDPs more effectively.

3. DEEP REINFORCEMENT LEARNING FOR CONTINUOUS CONTROL TASKS WITH STOCHASTIC POLICIES

3.1 Introduction

Continuous control tasks can be represented as MDPs with a limited number of relevant variables in a simulation environment that necessitate the operator's regural adjustment. This process consists of successive states and actions that stochastically affect the next state. These tasks differ from combinatorial problems in that they have a continuous search space, or the smoothness assumption assumes that similar actions apply to situations close to the input space, except for particular boundary requirements. (Abramson et al., 2003)

State-action pairs can be used to represent a task, with the states being tangible variables that define the physical process at a given point in time and in most cases, the desired action can be represented as a non-linear function of these states. An exact solution to those problems is difficult to compute.

3.2 Deep Reinforcement Learning

Deep reinforcement learning has recently gained popularity owing to its many successful real-world applications in robotics and games. Conventional RL policies, such as tabular, linear, and radial basis networks have a number of drawbacks, especially in terms of high-level behavior representation and the computational cost of updating parameters (Le et al., 2018). Combining RL and deep learning can extract more information from raw inputs by passing them through multiple neural layers, which can have a large number of parameters and represent highly nonlinear problems (Le et al., 2018). Using deep learning as a function approximator in RL (Aggarwal, 2018), deep RL methods can approximate optimal policies and/or value functions.

Numerous dynamic control problems involve a continuous state space, and in many cases, continuous action variables. Traditional reinforcement learning methods are not well adapted to these tasks because they assume discrete states and actions.

Discretization of continuous variables is an option. (Kimura, 2007). However, this method has several disadvantages. This can limit the generalization capabilities across similar situations or be prohibitively expensive, particularly in high-dimensional action spaces. Because of the curse of dimensionality, these lookup tables perform well in an environment with a finite number of states and actions; however, they are not appropriate for MDPs with continuous states and actions (Flet-Berliac, 2021). Function approximation is commonly utilized in the case of high-dimensional MDP because of its capacity to handle multi-dimensional continuous variables and generalize across similar states.

Value-based methods have remained under-explored deep RL methods in continuous control (Schaul et al., 2015), (Van Hasselt et al., 2016), (Wang et al., 2016). The learned value function in value-based approaches may contain multiple local maxima and saddle points if the action space is continuous. Consequently, policy-based techniques have traditionally been selected (Schulman et al., 2015). A neural network is typically employed as a function approximation for policies whose structure depend on observation and action spaces. Furthermore, because it is straightforward to sample and calculate its gradients, the Gaussian distribution is utilized as a stochastic policy in continuous control. The distinction between value- and policy-based methods is based on the trade-off between better performance and more complex strategies. However, value-based methods emphasize determining the maximum cumulative return and have a policy based on recommendations, which try to directly establish the optimal policy that maximizes the cumulative discounted reward. Furthermore, actor-critic approaches combine them and get better results for continuous control tasks with stochastic policies. DDPG (Deep Deterministic Policy Gradient) is the first actor-critic deep reinforcement learning method to address continuous control scenarios (Lillicrap et al., 2015). Subsequently, SVG (Stochastic Value Gradients) algorithm was suggested to deal with continuous control scenarios with stochastic policy gradients, allowing on-policy training (Heess et al., 2015).

In this study, we propose a novel actor-critic approach that combines the soft actor-critic method (Haarnoja et al., 2018) and supervised learning concepts to solve continuous control tasks with stochastic policies. We employ fully-factored Gaussian policies with the means of the action distributions as neural network outputs and the variances as separate trainable parameters, assuming that the action dimensions are independent.

3.2.1 Model uncertainty

In real-world problems, interaction with the physical world may be restricted for a variety of reasons such as safety and cost. The amount of interaction required in the real world can be reduced by learning the dynamic model of the environment. Furthermore, exploration can be performed on the learned models. With model-based approaches, the agent tries to learn the transition and reward functions that are utilized when taking an action. Gathering information on the dynamics of the environment is required to develop a model approach. Unlike model-based approaches, model-free approaches do not require this type of information. Model-free approaches directly illustrate the underlying MDP to learn an unknown model. Model-free policy search methods require more training data and longer training time than model-based methods (Kiran et al, 2021).

Many problems have unknown dynamics and rewards, therefore, the agent must learn to behave through experience. The agent chooses behaviors that optimize the long-term accumulation of rewards by observing state transitions and rewards as a result of its actions. Reinforcement learning is concerned with solving problems including model uncertainty. (Kochenderfer, 2015)

3.2.2 Twin delayed deep deterministic policy gradient (TD3)

TD3 (Fujimoto et al., 2018) is a more stable and effective variant of DDPG. A typical DDPG failure pattern is when the learnt Q-function starts to substantially overestimate Q-values, which causes the policy to break because the errors in the Q-function are exploited. (TD3) resolves this problem by proposing three critical tricks (OpenAI, 2021):

- TD3 uses two Q-functions instead of one to resolve Q-values overestimation, and uses the smaller of the two Q-values to form the targets in the Bellman error loss functions. (OpenAI, 2021)
- TD3 performs one policy update for every two Q-function updates. (OpenAI, 2021)

- TD3 adds noise to the target action, making it more difficult for the policy to exploit Q-function errors by smoothing out Q along changes in action. (OpenAI, 2021)

TD3, an off-policy algorithm, can only be used for environments with continuous action spaces.

3.2.3 Soft actor-critic (SAC)

SAC (Haarnoja et al., 2018a), which is defined for RL tasks involving continuous actions, is an off-policy optimization algorithm for stochastic policies. Entropy regularization is a key element of the SAC. The policy is trained to maximize the trade-off between the expected returns and entropy, which is a measure of the randomness of the policy (OpenAI, 2021). This is strongly related to the exploration-exploitation trade-off. More exploration occurs when the entropy increases. Therefore, it has the potential to accelerate learning. It can also prevent the policy from prematurely converging to a bad local optimum. The objective function of the SAC comprises both a reward term and an entropy term $\mathcal{H}$ weighted by α

$$J(\pi) = \sum_{t=0}^{T} \mathbb{E}_{(\mathbf{s}_t,\mathbf{a}_t)\sim\rho_\pi}\left[r(\mathrm{s}_t, \mathrm{a}_t) + \alpha\mathcal{H}\left(\pi(\cdot \mid \mathrm{s}_t)\right)\right] \tag{3.1}$$

SAC concurrently learns by using three networks: a state value function V parametrized by ψ, a soft Q-function Q parameterized by θ, and a policy function π parametrized by ϕ (Haarnoja et al., 2018).

The value network is trained to minimize the squared residual error (Haarnoja et al., 2018):

$$J_V(\psi) = \mathbb{E}_{\mathbf{s}_t\sim\mathcal{D}}\left[\frac{1}{2}\left(V_\psi(\mathrm{s}_t) - \mathbb{E}_{\mathbf{a}_t\sim\pi_\phi}\left[Q_\theta(\mathrm{s}_t, \mathrm{a}_t) - \log\pi_\phi(\mathrm{a}_t \mid \mathrm{s}_t)\right]\right)^2\right] \tag{3.2}$$

where $\mathcal{D}$ is the distribution of the replay buffers. The gradient of Eq. (3.2) can be estimated using an unbiased estimator (Haarnoja et al., 2018):

$$\hat{\nabla}_\psi J_V(\psi) = \nabla_\psi V_\psi(\mathrm{s}_t)\left(V_\psi(\mathrm{s}_t) - Q_\theta(\mathrm{s}_t, \mathrm{a}_t) + \log\,\pi_\phi(\mathrm{a}_t \mid \mathrm{s}_t)\right) \tag{3.3}$$

where the actions are sampled according to the current policy, instead of the replay buffer.

The Q network is trained by minimizing the soft Bellman residual (Haarnoja et al., 2018):

$$J_Q(\theta) = \mathbb{E}_{(\mathbf{s}_t,\mathbf{a}_t)\sim\mathcal{D}}\left[\frac{1}{2}\left(Q_\theta(\mathrm{s}_t,\mathrm{a}_t) - \hat{Q}(\mathrm{s}_t,\mathrm{a}_t)\right)^2\right] \tag{3.4}$$

with

$$\hat{Q}(\mathrm{s}_t,\mathrm{a}_t) = r(\mathrm{s}_t,\mathrm{a}_t) + \gamma\mathbb{E}_{\mathbf{s}_{t+1}\sim p}\left[V_{\bar{\psi}}(\mathrm{s}_{t+1})\right] \tag{3.5}$$

which again is optimized with stochastic gradients

$$\hat{\nabla}_\theta J_Q(\theta) = \nabla_\theta Q_\theta(\mathrm{a}_t,\mathrm{s}_t)\left(Q_\theta(\mathrm{s}_t,\mathrm{a}_t) - r(\mathrm{s}_t,\mathrm{a}_t) - \gamma V_{\bar{\psi}}(\mathrm{s}_{t+1})\right) \tag{3.6}$$

Lastly, the policy parameters are learned by minimizing the expected KL-divergence directly (Haarnoja et al., 2018):

$$J_\pi(\phi) = \mathbb{E}_{\mathbf{s}_t\sim\mathcal{D}}\left[\mathrm{D_{KL}}\left(\pi_\phi(\cdot\mid \mathrm{s}_t)\parallel \frac{\exp(Q_\theta(\mathrm{s}_t,\cdot))}{Z_\theta(\mathrm{s}_t)}\right)\right] \tag{3.7}$$

the policy is reparametrized by using a neural network transformation

$$\mathrm{a}_t = f_\phi(\epsilon_t;\mathrm{s}_t) \tag{3.8}$$

where ϵ_t is an input noise vector. The objective function in Equation 7 can be rewritten as follows:

$$J_\pi(\phi) = \mathbb{E}_{\mathbf{s}_t\sim\mathcal{D},\epsilon_t\sim\mathcal{N}}\left[\log\,\pi_\phi\big(f_\phi(\epsilon_t;\mathrm{s}_t)\mid \mathrm{s}_t\big) - Q_\theta\left(\mathrm{s}_t, f_\phi(\epsilon_t;\mathrm{s}_t)\right)\right] \tag{3.9}$$

The gradient of Equation 9 can be approximated with

$$\begin{aligned}\hat{\nabla}_\phi J_\pi(\phi) &= \nabla_\phi \log\,\pi_\phi(\mathrm{a}_t\mid \mathrm{s}_t)\\ &+ \left(\nabla_{\mathrm{a}_t}\log\,\pi_\phi(\mathrm{a}_t\mid \mathrm{s}_t) - \nabla_{\mathrm{a}_t}Q(\mathrm{s}_t,\mathrm{a}_t)\right)\nabla_\phi f_\phi(\epsilon_t;\mathrm{s}_t)\end{aligned} \tag{3.10}$$

where a_t is evaluated at $f_\phi(\epsilon_t;\mathrm{s}_t)$ (Haarnoja et al., 2018).

3.3 Guided Policy Search on Stochastic Policies

Guided policy search algorithms assist policy learning with additional information and mechanisms. The type of assistance can vary depending on the domain of the problem or the policy search method used.

Levine and Koltun (2013) proposed a guided policy search method using trajectory optimization to learn the policy directly and prevent poor local optima. Levine and Koltun (2013) employed an importance-sampled variation of the likelihood ratio estimator to incorporate off-policy guiding samples created using DP into the policy search. Guided policy search algorithms have been applied to continuous control domains such as challenging locomotion tasks, robotic manipulation, and control (Levine and Koltun, 2014), (Levine et al., 2015), (Levine et al., 2016). Levine and Abbeel (2014) performed a guided policy search under unknown dynamics using a hybrid method and optimized trajectory distributions for large continuous problems using iteratively refitted local linear models. Zhang et al. (2016a) proposed an approach to decompose the policy search problem into two phases, trajectory optimization and supervised learning, to acquire policies with effective memory and recall procedures in continuous control tasks. In the context of guided policy search, Zhang et al. (2016b) presented an approach that integrated model predictive control (MPC) and RL. To supervise the learning of the neural network policy, Zhang et al. (2016b) employed MPC, which only receives the full state of the system during the training phase. Zhang et al. (2016b) used a model-based technique to create guiding samples for guided policy search by combining MPC with offline trajectory optimization and assuming that a system dynamics approximation model was obtained in the training phase.

Guided RL techniques use additional state information throughout the learning process to improve their effectiveness in solving POMDPs. There are only a few prominent examples, such as the studies by (Levine and Koltun, 2013) and (Zhang et al., 2016b) on guided policy search methods.

3.4 Chapter Conclusion

In this chapter, we introduced how to use deep reinforcement learning algorithms to solve continuous control problems with stochastic policies and how to improve guided RL methods on stochastic policies to solve POMDPs more effectively. Guided policy search (GPS) approaches provide an idea for combining RL and supervised learning to solve POMDP problems by including previously shortened observation-action pairs into current state-action representations. We propose a new approach for solving continuous control POMDP problems by combining a model-free guided framework with SAC. Therefore, in this chapter, we provided preliminaries about Soft Actor-Critic (SAC) and guided policy search on stochastic policies. In the next chapter, we explain our proposed methodology, novel architecture, and interaction loop for SAC to solve partially observable continuous control tasks more efficientl

4. A GUIDED DEEP REINFORCEMENT LEARNING APPROACH FOR PARTIALLY OBSERVABLE MARKOV DECISION PROCESSES

4.1 Introduction

This chapter provides a detailed explanation of the proposed methodology, novel architecture, and interaction loop for Soft Actor-Critic (Haarnoja et al., 2018) to solve partially observable continuous control tasks more efficiently in much more exploration and interaction with the environment by using a guided policy search approach to optimize the policy.

4.2 Guided Soft Actor-Critic

Guided policy search (GPS) approaches (Levine and Koltun, 2013), (Levine and Koltun, 2014), (Levine and Abbeel, 2014), (Levine et al., 2015), (Levine et al., 2016), (Zhang et al., 2016a), (Zhang et al., 2016b) provide an idea for combining RL and supervised learning to solve POMDP problems by including previously shortened observation-action pairs into current state-action representations. Model-based approaches must learn a dynamic model of the environment to reduce the amount of interaction required in the real world. However, this can be challenging for complex tasks and requires extensive computations. On the other hand, model-free approaches learn a policy directly from interactions with the environment and have much more flexibility because they do not require an exact representation of the environment. To achieve effective outcomes using model-free approaches, an adequate number of interactions with the environment must be provided depending on the complexity of the task. Therefore, we focus on a model-free approach, as opposed to most traditional GPS methods. We present a new approach to solving continuous control POMDP problems by combining a model-free guided framework with SAC.

Our guided SAC, as in (Zhang et al., 2016b), has two phases: training and testing. In the training phase, the guided-SAC architecture consists of two actors and a critic. To generate samples stored in replay buffer $\mathcal{D}$, two actors interact with the same environment iteratively.

$$\mathcal{D} = \left(a_0^g, s_0^g, r_0^g, a_1^f, s_1^f, r_1^f, a_2^g, \ldots\right) \tag{4.1}$$

Figure 4.1 shows the guided-SAC architecture and the interaction loop of the guided SAC in the training phase.

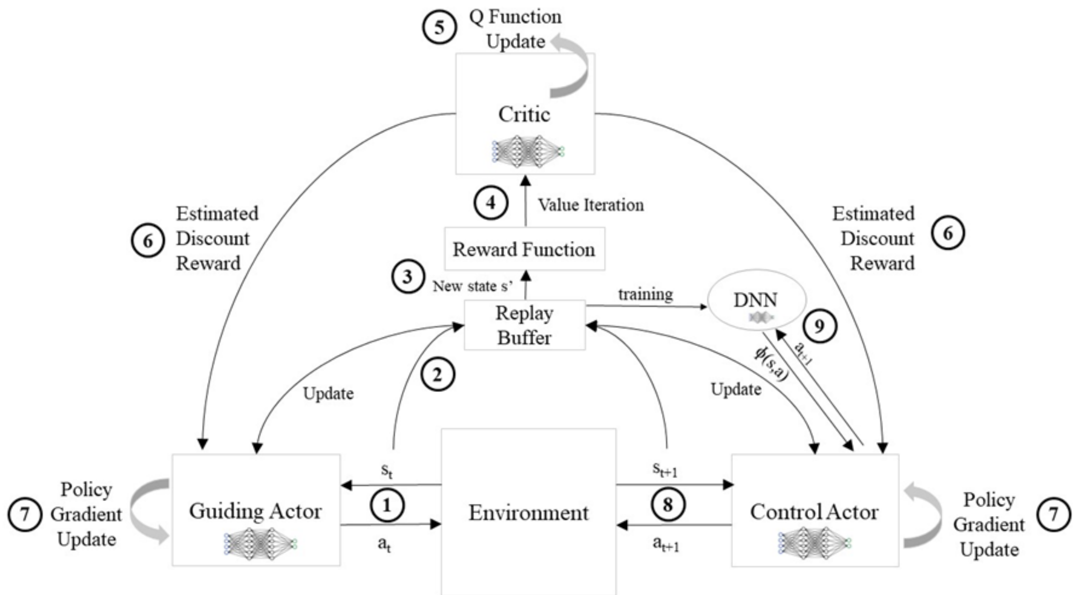


Figure 4.1 : Architecture and Interaction loop of Guided-SAC during the training phase. The numbers indicate the order of the steps.

The final control actor chooses actions based on partial observations, whereas the guiding actor chooses actions based on the full-state observations. The critic assesses the policy and, as a result, updates the action-value function by using data gathered through environmental observations. The targets for the Q functions are computed using:

$$y(r, s', d) = r + \gamma(1-d)\left(\min_{i=1,2} Q_{\phi_{\text{targ},i}}(s', \tilde{a}') - \alpha \log \pi_\theta(\tilde{a}' \mid s')\right), \tilde{a}' \sim \pi_\theta(\cdot \mid s') \tag{4.2}$$

and Q-functions are updated by one step of gradient descent using

$$\nabla_{\phi_i} \frac{1}{|B|} \sum_{(s,a,r,s',d)\in B} \left(Q_{\phi_i}(s,a) - y(r,s',d)\right)^2 \quad \text{for } i = 1,2 \tag{4.3}$$

The actors are then, fed the estimated discounted reward. Each policy gradient of the actors is updated using a one-step of gradient ascent, using

$$\nabla_{\theta f} \frac{1}{|B|} \sum_{s \in B} \left(\min_{i=1,2} Q_{\phi_i}(s, \tilde{a}_\theta(s)) - \alpha \log \pi_\theta(\tilde{a}_\theta(s) \mid s) \right) \tag{4.4}$$

$$\nabla_{\theta g} \frac{1}{|B|} \sum_{s \in B} \left(\min_{i=1,2} Q_{\phi_i}(s, \tilde{a}_\theta(s)) - \alpha \log \pi_\theta(\tilde{a}_\theta(s) \mid s) \right) \tag{4.5}$$

where $\tilde{a}_\theta(s)$ is a sample from $\pi_\theta(\cdot \mid s)$ which is differentiable w.r.t θ via the reparametrization trick.

The actor chooses an action according to the current state and interacts directly with the environment using the estimated action-value function. For each update, the guiding policy is trained with the history of observation-action pairs along with the current state from $\mathcal{D}$ and the control actor updates its policy with only the history of observation-action pairs from $\mathcal{D}$. Both are trained to maximize the expected reward while also maximizing entropy, as in the original concept of the SAC (Haarnoja et al., 2018a). Through α known as the entropy temperature, the agent chooses random actions to explore the environment. The gradient descent is used to adjust parameter to guarantee adequate exploration at the beginning of learning and, thereafter, a larger emphasis on maximizing the predicted reward. By decreasing the Kullback–Leibler divergence, the policy distribution for the Q function is adjusted towards the softmax distribution in the policy improvement step (Haarnoja et al., 2018b).

In our approach, to use the samples generated by various policies, the distributions of the two policies should be similar. We use the adjustable entropy temperature α becasue it improves the performance and stability of the method (Haarnoja et al., 2018b) and guarantees that they converge to the same policy.

$$\alpha \leftarrow \alpha - \lambda \hat{\nabla}_\alpha J(\alpha) \tag{4.6}$$

Update target networks with

$$\phi_{\text{targ},i} \leftarrow \rho \phi_{\text{targ},i} + (1 - \rho)\phi_i \text{ for } i = 1,2 \tag{4.7}$$

We also trained the DNN using the data stored in $\mathcal{D}$ to teach the control policy actor to act as a guiding policy actor. Using this iterative interaction approach, the same trajectory is generated by both the guiding and control policies.

Algorithm 1 presents the guided SAC approach. The numbers, which indicate the order of the steps in Figure 4.1, are referenced in the Algorithm 1.

Algorithm 1: Guided SAC

Input: initial policy parameters θ_f, θ_g, Q-function parameters ϕ_1, ϕ_2, empty replay buffer $\mathcal{D}$
Set target parameters equal to main parameters $\phi_{targ,1} \leftarrow \phi_1$, $\phi_{targ,2} \leftarrow \phi_2$
while
Run the guiding policy network $\pi_{\theta g}(a_t \mid h_t, s_t)$ and the final control policy $\pi_{\theta g}(a_t \mid h_t)$ iteratively (Steps 1, 8 in Fig. 4.1)
Store in replay buffer $\mathcal{D}$ with eq. (4.1) (Step 2 in Fig. 4.1)
If s' is terminal, reset environment state.
if it'**s** time to update **then**
for j in range(however many updates) **do**
Randomly sample a batch of transitions, $B = \{(s, a, r, s', d)\}$ from $\mathcal{D}$ (Steps 3, 4 in Fig. 4.1)
Compute targets for the Q functions with eq. (4.2) (Step 5 in Fig. 4.1)
Update Q-functions by one step of gradient descent using eq. (4.3) (Step 5 in Fig. 4.1)
Update policy by one step of gradient ascent using eq. (4.4), (4.5) (Step 6 in Fig. 4.1)
Adjust entropy temperature with eq. (16)
Update target networks with eq. (17) (Step 7 in Fig. 4.1)
end for
end if
until convergence

During the test phase, only the final control actor interacts with the environment. (Figure 4.2)

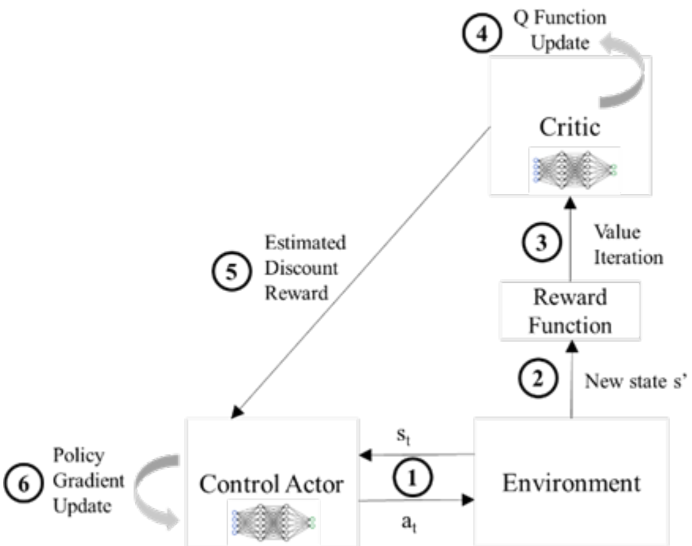


Figure 4.2 : Interaction loop of guided-SAC during the test phase. The numbers indicate the order of the steps.

4.3 Chapter Conclusion

To summarize our contribution, we propose a novel approach called guided SAC that combines a model-free guided framework with SAC for POMDPs. We proposed a new agent-environment interaction method for generating training samples in which the guiding and control actors interact with the same environment iteratively. The control actor selects actions based on partial observations, whereas the guiding actor can take actions by perceiving full-state observations. In our approach, to use the samples generated by various policies, the distributions of the two policies should be similar. We further introduced an additional update step to use the adjustable entropy temperature α, as it improves the performance and stability of the method and guarantees that they converge to the same policy.

5. EXPERIMENTS ON PARTIALLY OBSERVABLE ROBOTIC TASKS CONSTRUCTED BASED ON CONTINUOUS CONTROL PROBLEMS

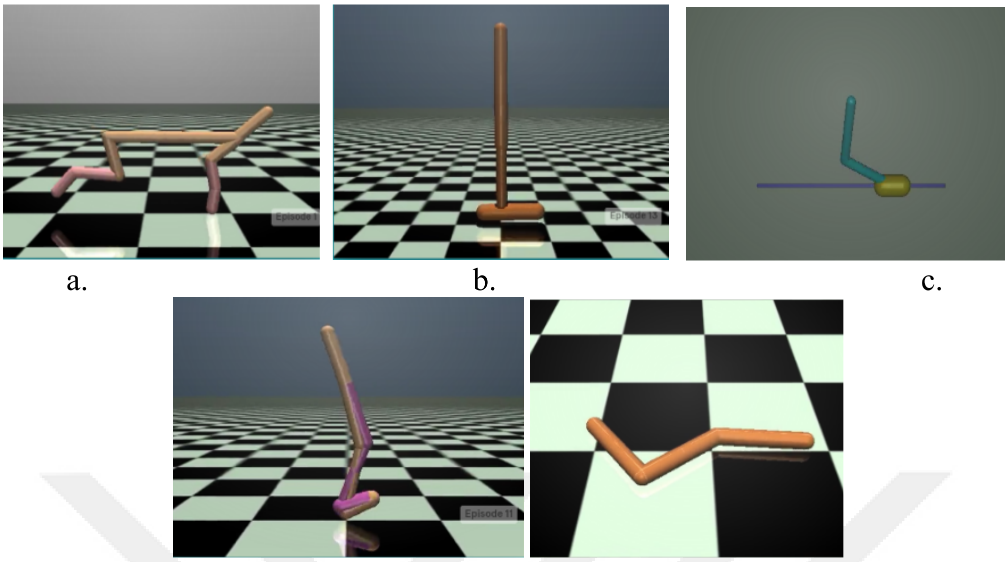
5.1 Introduction

Deep reinforcement learning necessitates a great deal of trial-and-error episodes in the real world to learn an optimal policy. Simulators are needed to accomplish outcomes in a cost-effective and timely manner because they allow these episodes to occur in the digital world. As an example, consider teaching a robot to walk by observing a real physical robot attempt and fail 100,000 times until it can reliably and consistently walk. To avoid causing damage in the real world, it is preferable to first train in a simulation platform before transferring it to the real world. We ran experiments in five different MuJoCo (Todorov et al., 2012) environments that provide continuous control tasks of increasing difficulty and action dimensions.

In this chapter, we investigated the following research question during the experiments: Can our proposed guided deep reinforcement learning approach outperform the original DRL baseline algorithms on partially observable robotic tasks constructed based on continuous control problems? We selected the state-of-the-art DRL baseline algorithms TD3 (Fujimoto et al., 2018) and SAC (Haarnoja et al., 2018a) to compare the performance of the proposed guided SAC approach.

5.2 Environments and Tasks

In this study, we compared our guided SAC approach with the state-of-the-art baseline algorithms (SAC and TD3) on five partially observable tasks (HalfCheetah-v2, Hopper-v2, InvertedDoublePendulum-v2, Walker2d-v2, and Swimmer-v2) that we constructed based on high-dimensional, continuous control problems in MuJoCo in the OpenAI Gym framework. (Figure 5.1) Typical observations for these tasks include the positions, angles, velocities or forces. Based on the original tasks, have a fully observable state-space, we built a modification to simulate partial observability in real applications. The modification details of the Mujoco tasks for POMDP are given in Appendix A.



a. b. c.

d. e.

Figure 5.1 : POMDP Mujoco Tasks – a. HalfCheetah-v2 b. Hopper-v2 c. InvertedDoublePendulum-v2 d. Walker2d-v2 e. Swimmer-v2.

5.3 Experimental Settings

To compare our results, the original implementations of SAC and TD3 as well as the OpenAI baselines were used. We trained the algorithms for a total of 1-million-time steps on MuJoCo tasks. One core on an Intel® Xeon® Gold 6148 Processor and 16 GB of RAM were used for the computation of each job. Each task lasted approximately 12 hours. (Training time of each task for 1 million times steps are given in Appendix B) All trials were performed with a total of ten random seeds in each. Running the mean policy without action noise for ten trajectories once per 10,000 steps was used to assess the performance, and the average return over ten trajectories was reported. The results of our experiments were plotted as learning curves for the average cumulative reward over the last ten episodes. Feedforward neural networks are used to represent policies and/or value functions in all the evaluation techniques. The SAC and TD3 algorithms use two hidden neural network layers of size (256,256) with rectified linear units (ReLU) as the activation function (Fujimoto et al., 2018), (Haarnoja et al., 2018a). For the baseline algorithms, the default hyperparameters from OpenAI SpinningUp (OpenAI, 2021) were used. The hyperparameters for our proposed algorithm are given in Appendix C. The observation history has a fixed length of four, which means that the input for the RL algorithm includes the current observation as well as the four previous observations.

5.4 Results

To compare our guided SAC approach with SAC and TD3, we ran all the tasks for full observability and POMDP. The learning curves for the sampled tasks are shown in Figure 5, with the first column representing the fully observable performance and the second column representing POMDP findings. Tables 2 and 3 summarize the results.

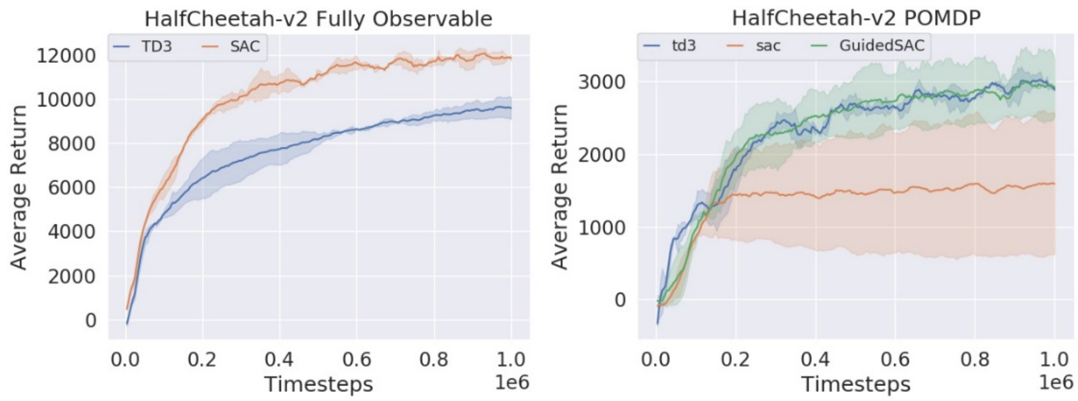
Table 5.1 : Maximum Average Return on Fully Observable.

Task	SAC	TD3
HalfCheetah-v2	**12610.32±120.89**	10050.08±127.51
Hopper-v2	3650.64±3.4	**3786.79±4.95**
InvertedDoublePendulum-v2	**9359.85**	9359.66
Walker2d-v2	4470.69±78.31	**5857.5± 34.78**
Swimmer-v2	45.95±1.69	**55.16±9.36**

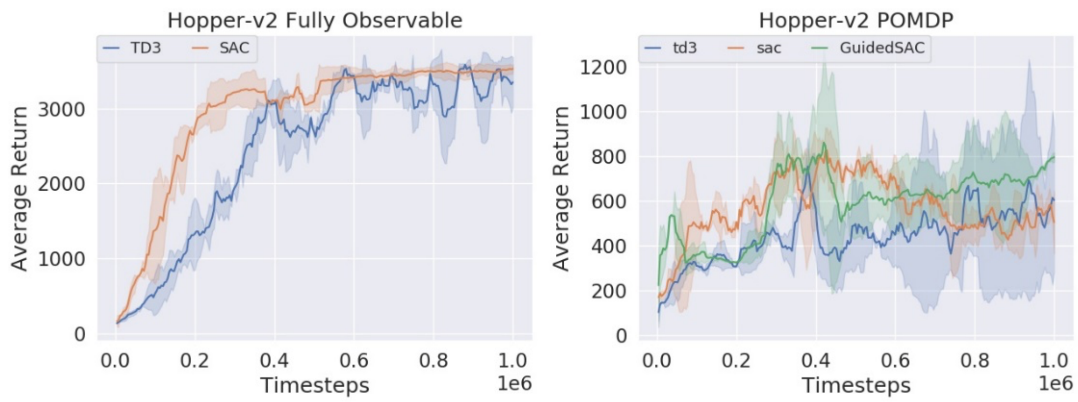
Table 5.2 : Maximum Average Return on POMDP Mujoco Tasks.

Task	SAC	TD3	Guided SAC	Δ
HalfCheetah-v2	2508.82±148.46	3431.44±193.51	**3658.26±123.59**	7%
Hopper-v2	1444.73±21	1586.35±460	**1784.8±11.45**	10%
InvertedDouble Pendulum-v2	6313.95±3763	**9306±1.16**	9120.46±688.6	-2%
Walker2d-v2	2269.17±431.78	897±172.52	**2616.89±434.68**	15%
Swimmer-v2	32.19±5.3	34.42±2.15	**41.46±1.83**	20%

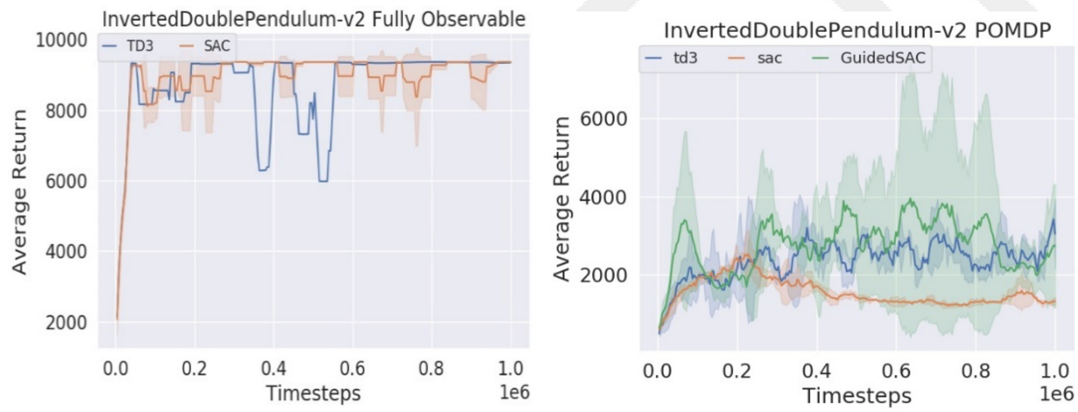
The guided SAC performed best on the four POMDP MuJoCo tasks. The guided SAC outperformed the state-of-the-art DRL baseline algorithm by more than 7~20% for each task. TD3 could perform better than guided SAC on POMDP InvertedDoublePendulum-v2 Task. In general, guided SAC achieves significantly better performance on difficult POMDP tasks than baselines.



a. HalfCheetah-v2

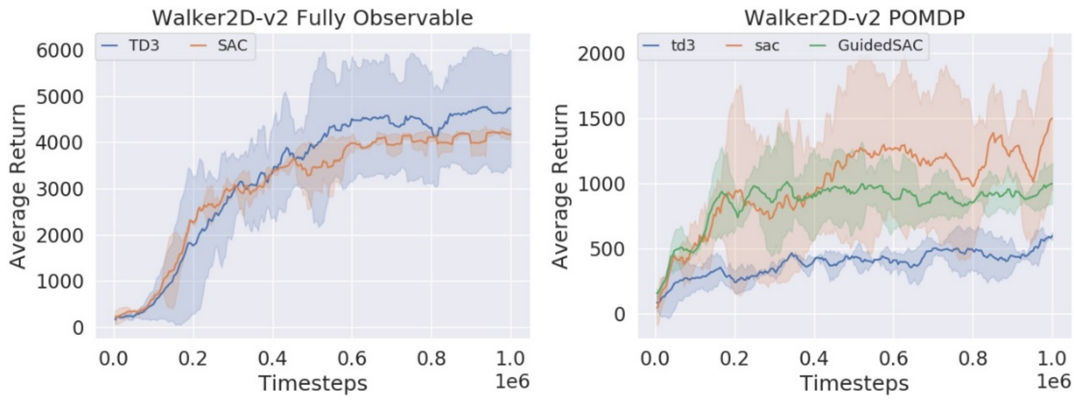


b. Hopper-v2

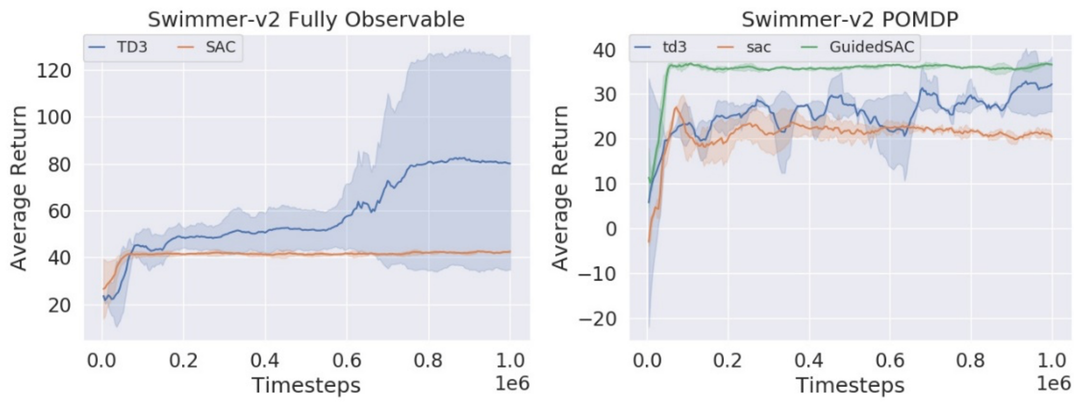


c. InvertedDoublePendulum-v2

Figure 5.2 : Learning curves for both fully observable and POMDP MuJoCo tasks. Learning curves for the average cumulative reward over the last 10 episodes for the sampled tasks, with the first column representing fully observable performance and the second column representing POMDP findings. The algorithms were executed for ten trajectories once per 10,000 steps in each iteration.



d. Walker2D-v2



e. Swimmer-v2

Figure 5.2 (continued): Learning curves for both fully observable and POMDP MuJoCo tasks. Learning curves for the average cumulative reward over the last 10 episodes for the sampled tasks, with the first column representing fully observable performance and the second column representing POMDP findings. The algorithms were executed for ten trajectories once per 10,000 steps in each iteration.

5.5 Chapter Conclusion

In this chapter, we evaluated our approach using modified, partially observable MuJoCo tasks. The proposed guided SAC was compared to the state-of-the-art DRL baseline algorithms (SAC and TD3) on the POMDP versions of the sampled continuous control tasks in MuJoCo. We learned the optimal policy successfully and obtained good empirical results, showing that the guided SAC outperforms the baselines on POMDPs. Our guided soft actor-critic approach outperformed the state-of-the-art DRL baseline algorithm, achieving a maximum average return of 7~20% on four out of five partially observable tasks.

6. EXPERIMENTS ON DECISION MAKING UNDER UNCERTAINTY FOR AUTONOMOUS DRIVING

6.1 Introduction

Autonomous driving is a difficult problem to solve in the real world, where states and actions are in a high-dimensional space. The behavior of pedestrians or other vehicles on the road and in weather conditions involves high uncertainty. The behavior of pedestrians is difficult to predict and the uncertainty in the behavior of other traffic elements can also cause accidents.

In the literature, four studies used deep reinforcement learning and formulated autonomous driving as a POMDP. Wang and Chan (2017) proposed a deep reinforcement learning solution with a LSTM architecture to learn autonomous driving policies under uncertainty in an on-ramp merge scenario. The LSTM architecture was used to combine the impact of historical and actual driving behaviors on decision-making. In the DQN, The Q-function approximator was utilized to determine actions based on additional historical knowledge using the internal state from the LSTM as input. To deal with the local optimum and instability difficulties, a second Q-network, by updating the Q-network parameters using a replay buffer, was deployed. Qiao et al. (2018) extended the LSTM solution proposed by Isele et al. (2018) to produce continuous actions rather than discrete ones by defining the autonomous driving as a POMDP. Pusse and Klusch (2019) developed a hybrid solution that combines the advantages of NavA3C (Mirowski et al., 2016) and the IS-DESPOT (Luo et al., 2019) approach, which is used to analytically solve POMDPs online, to perform accident-free navigation of the self-driving car until it reaches the destination. In this study, experimental evaluation results were discussed for simulated single and multiple pedestrian-car accident scenarios based on the German In-Depth Accident Study (GIDAS, 2022).

This chapter focuses on situations in which the self-driving car partially or completely loses its ability to observe the environment. In this chapter, a new approach is proposed to solve the autonomous driving problem depending on decision-making under uncertainty as a POMDP using the guided SAC method. The experiments were carried out in MATLAB, and the self-driving car was trained and tested for a scenario in which it encountered a pedestrian crossing the road. According to the test results, the controller exhibits the desired control behavior and avoids collisions, making no mistakes in various uncertain environments.

6.2 Problem Statement

For partially observable autonomous driving tasks, the self-driving car is expected to reach its destination without colliding with any static obstacles or other dynamic traffic users (vehicles and pedestrians). While the behavior of dynamic objects is time-varying and unpredictable, a self-driving car needs to decide actions with partial observations, which increases the difficulty level of the problem. Therefore, it is necessary to evaluate longer historical data to make the most appropriate decisions for the current situation under uncertainty. Otherwise, pedestrians may completely lose their information completely when the vehicle enters the "blind" area.

The self-driving car decision-making can be modeled as a POMDP, which a self-driving car can solve online in real time.

POMDP is defined by the group ($S, \mathcal{A}, O, T, \mathcal{O}, R$, γ) where S state space, $\mathcal{A}$ action space, O observation space, T transition model, R reward function, $\mathcal{O}$ observation model and γ discount factor.

In this study, it is assumed that a self-driving car has a 360-degree perception of the environment and that pedestrians can be observed at a viewing distance up to 50 meters if there are no obstacles, and interact only with pedestrians as a traffic user. The state space is assumed to be continuous and the base states for self-driving car and pedestrian are determined as the position (x, y), speed υ, and direction θ (Figure 6.1).

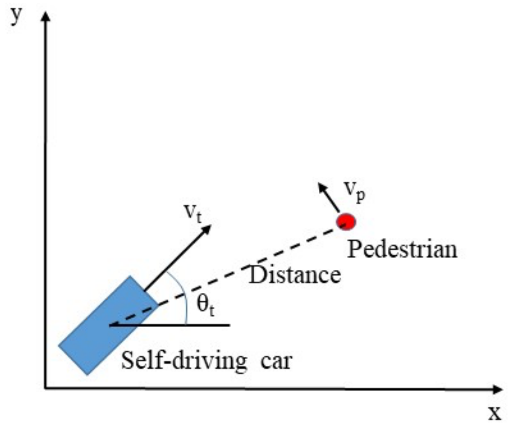


Figure 6.1 : Basic states for self-driving car and pedestrian.

Given that these situations for pedestrians depend on their uncertain behavior (b), the output of the behavior prediction is added as an extra case for these vehicles.

$$S = (s_{oto}, s_1, \ldots, s_n)_t \tag{6.1}$$

$$S_{oto} = (x, y, \upsilon, \theta)_t \text{ , } s_i = (x, y, \upsilon, \theta, b)_t \tag{6.2}$$

An autonomous driving policy must control several different actuators. In this study, continuous value actuators for self-driving car control are considered for the steering angle and acceleration.

$$\mathcal{A} = (\alpha, acceleration) \tag{6.3}$$

The acceleration action set includes three different actions:

acceleration = {Accelerator, Steady, Brake}

The transition pattern is defined as $T: S \times \mathcal{A} \times S \rightarrow [0,1]$. In this model, the probability of going to the next $\mathbf{s}'$ state after the action a while in state s is $T(\mathbf{s}, \mathbf{a}, \mathbf{s}')$. The observation includes measurements of the self-driving car's condition and the observable portion of pedestrians detected in the scene. The observation set is defined as:

$$O = (o_{oto}, o_1, \ldots, o_n)_t \tag{6.4}$$

$$o_{oto} = (x, y, \upsilon, \theta)_t \text{ , } o_i = (x, y, \upsilon, \theta, b)_t \tag{6.5}$$

The observation probability function $O(\mathbf{s}', \mathbf{a}, \mathbf{o})$ returns **o,** the probability of observation, when the state $\mathbf{s}'$is passed after the action **a**. This function models the noise in the sensors of a self-driving car and the uncertainty in the movement of pedestrians.

The reward function $r(\mathbf{s}, \mathbf{a})$ provides a positive or negative reward depending on the current conditions of the environment. Rewards:

- If a pedestrian crosses safely, the self-driving car receives a positive reward r_{target}.
- The self-driving car receives a negative reward r_{near}, if the distance to the pedestrian is between the safe distance and collision distance.
- The self-driving car receives negative reward r_{crash}, if it hits the pedestrian

The discount factor γ was selected as 0.98.

6.3 Experimental Settings

In this study, a simulation environment was set up using MATLAB/SIMULINK. In this simulation environment, scenarios in which a self-driving car encounters a pedestrian crossing the road are created. The driving behavior of a self-driving car was modeled using a kinematic bicycle model. (Kong et al., 2015).

The starting position of the self-driving car was assumed to be fixed in each harness section. Pedestrian speed during the training phase was determined randomly with a uniform distribution of 0.5 and 4 m/s, including pedestrian behaviors at different speeds (slow, normal, and fast). The initial speed of the self-driving car was determined randomly, with a uniform distribution of 20 km/h and 50 km/h, which is the maximum speed limit in a residential area in Turkey. The initial distance between the pedestrian and self-driving car was determined randomly, with a uniform distribution of 10 and 70 m. In the training and testing phase, for the detection of near-miss and collision situations, a distance of over 3 meters from the center of the self-driving car to the pedestrian was accepted as a safe distance, between 1.5 and 3 meters as a near-miss, and below 1.5 meters as a collision.

In this study, it was assumed that pedestrians can be observed at a viewing distance up to 50 meters if there are no obstacles. When the pedestrian is walking on the shadowed portion of the road, the self driving car cannot observe the pedestrian's behavior in the real world. In our simulation environment, the pedestrian was generated randomly from one of the 10 spawner points, all of which were behind the parked cars on both sides of the road. Thus, it was ensured that the environment was partially observable.

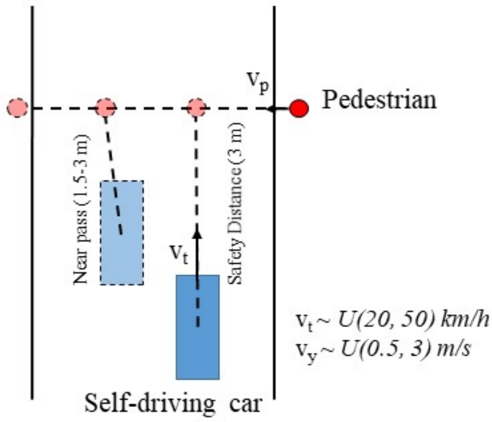


Figure 6.2 : Self-driving car-pedestrian encounter scenario.

During the training phase, training was performed with a total of 10000 section runs using one core and 16 GB RAM on the Intel® Xeon® Gold 6148 Processor. The hyperparameters for our proposed algorithm are given in App. A.

6.4 Results

To better understand the performance of the guided SAC solution for partially observable states, both partially observable and fully observable state were subjected to trainings and performance tests.

Performance tests were carried out with 1000 different harnesses where the starting speeds of the self-driving car and pedestrian were different. Table 6.1 shows the average performance of the self-driving car for 1000 trials, when the pedestrian crossing the road was fully observable and partially observable under the same conditions.

Table 6.1 : Results.

Environment State	Crash (%)	Impact Speed (km/h)	Near Pass (%)
Fully Observable	1.2	20.76	2.3
Partially Observable	1.8	24.32	3.9

The guided SAC approach performed close to the full observable performance in partially observable situations in terms of crash and near pass. The impact velocity at the time of impact was greater because of the delayed response compared with the fully observable states.

6.5 Chapter Conclusion

In this chapter, a new approach is introduced to solve the autonomous driving problem, which focuses on decision making under uncertainty, as a POMDP, using the guided SAC method. The training was carried out by creating scenarios where the starting speeds of the self-driving car and the pedestrian and the starting distance to each other were different, and the self-driving car encountered a pedestrian crossing the road. To compare the performance of the guided SAC solution against partially observable cases with fully observable cases, performance tests were performed with 1000 different harnesses, and the results are presented. The results showed that the controller exhibited the desired control behavior and had near-full observable performance in various uncertainty situations.

7. CONCLUSIONS AND RECOMMENDATIONS

To summarize our contribution, we proposed a novel approach called guided SAC that combines a model-free guided framework with SAC for POMDPs. We proposed a new agent-environment interaction method for generating training samples in which the guiding and control actors interact with the same environment iteratively. The control actor selects actions based on partial observations, whereas the guiding actor can take action by perceiving full-state observations. We evaluated our approach using modified partially observable MuJoCo tasks. The proposed guided SAC was compared to the state-of-the-art DRL baseline algorithms (SAC and TD3) on the POMDP versions of the sampled continuous control tasks in MuJoCo. We learned the optimal policy successfully and obtained good empirical results, showing that the guided SAC outperforms the baselines on POMDPs. Our guided soft actor-critic approach outperformed the state-of-the-art DRL baseline algorithm, gaining 7~20% more maximum average return on four out of five partially observable tasks.

To solve the autonomous driving problem, we focused on decision making under uncertainty as a POMDP, using the guided SAC method. The training was carried out by creating scenarios where the starting speeds of the self-driving car and the pedestrian and the starting distance to each other were different, and the self-driving car encountered a pedestrian crossing the road. To compare the performance of the guided SAC solution against partially observable cases with fully observable cases, performance tests were performed with 1000 different harnesses, and the results are presented. The results showed that the controller exhibited the desired control behavior and had a near-full observable performance in various uncertainty situations.

Even we focused on POMDPs, extending our approach for mixed observable Markov decision processes (MOMDP) is straightforward. In future studies, we may apply our proposed approach to the tasks, formulated as a MOMDP.

Also, in future studies, we aim to apply our proposed algorithm to other autonomous driving use cases for partially observable environments, such as ramp merging, intersection.

REFERENCES

Abramson, M., Pachowicz, P., & Wechsler, H. (2003). Competitive reinforcement learning in continuous control tasks. In *Proceedings of the International Joint Conference on Neural Networks, 2003.* (Vol. 3, pp. 1909-1914). IEEE.

Aggarwal, C. C. (2018). *Neural networks and deep learning*. Springer, 10, 978-3.

Andrychowicz, M., Wolski, F., Ray, A., Schneider, J., Fong, R., Welinder, P., ... & Zaremba, W. (2017). Hindsight experience replay. Advances in neural information processing systems, 30.

Azizzadenesheli, K., Lazaric, A., & Anandkumar, A. (2017). Experimental results: Reinforcement learning of pomdps using spectral methods. arXiv preprint arXiv:1705.02553.

Azizzadenesheli, K., Bera, M. K., & Anandkumar, A. (2018). Trust region policy optimization for pomdps.

Azizzadenesheli, K., Yue, Y., & Anandkumar, A. (2018). Policy gradient in partially observable environments: Approximation and convergence. arXiv preprint arXiv:1810.07900.

Babaeizadeh, M., Frosio, I., Tyree, S., Clemons, J., & Kautz, J. (2016). Reinforcement learning through asynchronous advantage actor-critic on a gpu. arXiv preprint arXiv:1611.06256.

Bellemare, M. G., Dabney, W., & Munos, R. (2017). A distributional perspective on reinforcement learning. In International Conference on Machine Learning (pp. 449-458). PMLR.

Clemente, A. V., Castejón, H. N., & Chandra, A. (2017). Efficient parallel methods for deep reinforcement learning. arXiv preprint arXiv:1705.04862.

Dabney, W., Rowland, M., Bellemare, M., & Munos, R. (2018). Distributional reinforcement learning with quantile regression. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 32, No. 1).

Dabney, W., Ostrovski, G., Silver, D., & Munos, R. (2018). Implicit quantile networks for distributional reinforcement learning. In International conference on machine learning (pp. 1096-1105). PMLR.

Dai, B., Shaw, A., He, N., Li, L., & Song, L. (2017). Boosting the actor with dual critic. arXiv preprint arXiv:1712.10282.

Diederich, A. (2001). Sequential Decision Making. *International encyclopedia of the social & behavioral sciences*. Amsterdam: Elsevier.

Doan, T., Mazoure, B., & Lyle, C. (2018). Gan q-learning. arXiv preprint arXiv:1805.04874.

Egorov, M. (2015). *Deep reinforcement learning with pomdps*. Tech. Rep.(Technical Report, Stanford University, 2015).

Espeholt, L., Soyer, H., Munos, R., Simonyan, K., Mnih, V., Ward, T., ... & Kavukcuoglu, K. (2018). Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. In International Conference on Machine Learning (pp. 1407-1416). PMLR.

Foerster, J. N., Assael, Y. M., de Freitas, N., & Whiteson, S. (2016). Learning to communicate to solve riddles with deep distributed recurrent q-networks. arXiv preprint arXiv:1602.02672.

Fujimoto, S., Hoof, H., & Meger, D. (2018). Addressing function approximation error in actor-critic methods. In International conference on machine learning (pp. 1587-1596). PMLR.

Farquhar, G., Rocktäschel, T., Igl, M., & Whiteson, S. (2017). Treeqn and atreec: Differentiable tree-structured models for deep reinforcement learning. arXiv preprint arXiv:1710.11417.

Flet-Berliac, Y. (2021). *Sample-efficient deep reinforcement learning for control, exploration and safety* (Doctoral dissertation, Université de Lille).

François-Lavet, V., Bengio, Y., Precup, D., & Pineau, J. (2019). Combined reinforcement learning via abstract representations. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 33, No. 01, pp. 3582-3589).

Fujimoto, S., Hoof, H., & Meger, D. (2018). Addressing function approximation error in actor-critic methods. In International conference on machine learning (pp. 1587-1596). PMLR.

GIDAS (2022), The German In-Depth Accident Study, https://www.gidas.org/

Gruslys, A., Dabney, W., Azar, M. G., Piot, B., Bellemare, M., & Munos, R. (2017). The reactor: A fast and sample-efficient actor-critic agent for reinforcement learning. arXiv preprint arXiv:1704.04651.

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

Haarnoja, T., Tang, H., Abbeel, P., & Levine, S. (2017). Reinforcement learning with deep energy-based policies. In International Conference on Machine Learning (pp. 1352-1361). PMLR.

Haarnoja, T., Zhou, A., Abbeel, P., & Levine, S. (2018a). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In International conference on machine learning (pp. 1861-1870). PMLR.

Haarnoja, T., Zhou, A., Hartikainen, K., Tucker, G., Ha, S., Tan, J., ... & Levine, S. (2018b). Soft actor-critic algorithms and applications. arXiv preprint arXiv:1812.05905.

Hausknecht, M., & Stone, P. (2015). Deep recurrent q-learning for partially observable mdps. In 2015 AAAI fall symposium series.

Heess, N., Hunt, J. J., Lillicrap, T. P., & Silver, D. (2015). Memory-based control with recurrent neural networks. arXiv preprint arXiv:1512.04455.

Hessel, M., Modayil, J., Van Hasselt, H., Schaul, T., Ostrovski, G., Dabney, W., ... & Silver, D. (2018). Rainbow: Combining improvements in deep reinforcement learning. In Thirty-second AAAI conference on artificial intelligence.

Horgan, D., Quan, J., Budden, D., Barth-Maron, G., Hessel, M., Van Hasselt, H., & Silver, D. (2018). Distributed prioritized experience replay. arXiv preprint arXiv:1803.00933.

Igl, M., Zintgraf, L., Le, T. A., Wood, F., & Whiteson, S. (2018). Deep variational reinforcement learning for POMDPs. In International Conference on Machine Learning (pp. 2117-2126). PMLR.

Isele, D., Rahimi, R., Cosgun, A., Subramanian, K., & Fujimura, K. (2018). Navigating occluded intersections with autonomous vehicles using deep reinforcement learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 2034-2039). IEEE.

Jin, P., Keutzer, K., & Levine, S. (2018). Regret minimization for partially observable deep reinforcement learning. In International conference on machine learning (pp. 2342-2351). PMLR.

Kardell, S., & Kuosku, M. (2017). *Autonomous vehicle control via deep reinforcement learning* (Master's thesis).

Karkus, P., Hsu, D., & Lee, W. S. (2017). Qmdp-net: Deep learning for planning under partial observability. Advances in neural information processing systems, 30.

Kiran, B. R., Sobh, I., Talpaert, V., Mannion, P., Al Sallab, A. A., Yogamani, S., & Pérez, P. (2021). Deep reinforcement learning for autonomous driving: A survey. IEEE Transactions on Intelligent Transportation Systems.

Kochenderfer, M. J. (2015). *Decision making under uncertainty: theory and application*. MIT press.

Kong, J., Pfeiffer, M., Schildbach, G., & Borrelli, F. (2015). Kinematic and dynamic vehicle models for autonomous driving control design. In *2015 IEEE intelligent vehicles symposium (IV)* (pp. 1094-1099). IEEE.

Lee, H. Deep Reinforcement Learning With Forward Prediction, Memory, and Hierarchy. Accessed: Nov. 2, 2021. [Online]. Available: https://rll.berkeley.edu/deeprlcoursesp17/docs/honglak_lecture.pdf

Levine, S., & Koltun, V. (2013). Guided policy search. In *International conference on machine learning* (pp. 1-9). PMLR.

Levine, S., & Koltun, V. (2014). Learning complex neural network policies with trajectory optimization. In *International Conference on Machine Learning* (pp. 829-837). PMLR.

Levine, S., Wagener, N., & Abbeel, P. (2015). Learning contact-rich manipulation skills with guided policy search (2015). *arXiv preprint arXiv:1501.05611*.

Levine, S., Finn, C., Darrell, T., & Abbeel, P. (2016). End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, *17*(1), 1334-1373.

Levine, S., & Abbeel, P. (2014). Learning neural network policies with guided policy search under unknown dynamics. *Advances in neural information processing systems*, *27*.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., ... & Wierstra, D. (2015). Continuous control with deep reinforcement learning. arXiv preprint arXiv:1509.02971.

Lin, K., Zhao, R., Xu, Z., & Zhou, J. (2018). Efficient large-scale fleet management via multi-agent deep reinforcement learning. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (pp. 1774-1783).

Liu, B., Liu, J., Ghavamzadeh, M., Mahadevan, S., & Petrik, M. (2016). Proximal Gradient Temporal Difference Learning Algorithms. In IJCAI (pp. 4195-4199).

Luo, Y., Bai, H., Hsu, D., & Lee, W. S. (2019). Importance sampling for online planning under uncertainty. *The International Journal of Robotics Research*, *38*(2-3), 162-181.

Le, T. P., Vien, N. A., & Chung, T. (2018). A deep hierarchical reinforcement learning algorithm in partially observable Markov decision processes. Ieee Access, 6, 49089-49102.

McAllister, R., & Rasmussen, C. E. (2017). Data-efficient reinforcement learning in continuous state-action Gaussian-POMDPs. Advances in Neural Information Processing Systems, 30.

Mirowski, P., Pascanu, R., Viola, F., Soyer, H., Ballard, A. J., Banino, A., ... & Hadsell, R. (2016). Learning to navigate in complex environments. *arXiv preprint arXiv:1611.03673*.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *nature*, *518*(7540), 529-533.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., ... & Kavukcuoglu, K. (2016). Asynchronous methods for deep reinforcement learning. In International conference on machine learning (pp. 1928-1937). PMLR.

Morgenstern, O., & Von Neumann, J. (1953). *Theory of games and economic behavior*. Princeton university press.

Munos, R., Stepleton, T., Harutyunyan, A., & Bellemare, M. (2016). Safe and efficient off-policy reinforcement learning. Advances in neural information processing systems, 29.

Nachum, O., Norouzi, M., & Schuurmans, D. (2016). Improving policy gradient by exploring under-appreciated rewards. arXiv preprint arXiv:1611.09321.

Nachum, O., Norouzi, M., Xu, K., & Schuurmans, D. (2017). Bridging the gap between value and policy based reinforcement learning. Advances in neural information processing systems, 30.

Nachum, O., Norouzi, M., Xu, K., & Schuurmans, D. (2017). Trust-pcl: An off-policy trust region method for continuous control. arXiv preprint arXiv:1707.01891.

Nachum, O., Gu, S. S., Lee, H., & Levine, S. (2018). Data-efficient hierarchical reinforcement learning. Advances in neural information processing systems, 31.

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 2018 IEEE International Conference on Robotics and Automation (ICRA) (pp. 7559-7566). IEEE.

O'Donoghue, B., Munos, R., Kavukcuoglu, K., & Mnih, V. (2016). Combining policy gradient and Q-learning. arXiv preprint arXiv:1611.01626.

Oh, J., Singh, S., & Lee, H. (2017). Value prediction network. Advances in neural information processing systems, 30.

Oh, J., Chockalingam, V., & Lee, H. (2016). Control of memory, active perception, and action in minecraft. In International Conference on Machine Learning (pp. 2790-2799). PMLR.

OpenAI. Spinning up. Accessed: Nov. 2, 2021. [Online]. Available: https://spinningup.openai.com/

Pajarinen, J., Thai, H. L., Akrour, R., Peters, J., & Neumann, G. (2019). Compatible natural gradient policy search. Machine Learning, 108(8), 1443-1466.

Pusse, F., & Klusch, M. (2019). Hybrid online pomdp planning and deep reinforcement learning for safer self-driving cars. In *2019 IEEE Intelligent Vehicles Symposium (IV)* (pp. 1013-1020). IEEE.

Puterman, M. L. (1990). Markov decision processes. Handbooks in operations research and management science, 2, 331-434.

Puterman, M. L. (2014). *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

Qiao, Z., Muelling, K., Dolan, J., Palanisamy, P., & Mudalige, P. (2018). Pomdp and hierarchical options mdp with continuous actions for autonomous driving at intersections. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)* (pp. 2377-2382). IEEE.

Racanière, S., Weber, T., Reichert, D., Buesing, L., Guez, A., Jimenez Rezende, D., ... & Wierstra, D. (2017). Imagination-augmented agents for deep reinforcement learning. Advances in neural information processing systems, 30.

Schaul, T., Horgan, D., Gregor, K., & Silver, D. (2015). Universal value function approximators. In International conference on machine learning (pp. 1312-1320). PMLR.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., & Moritz, P. (2015). Trust region policy optimization. In International conference on machine learning (pp. 1889-1897). PMLR.

Schulman, J., Moritz, P., Levine, S., Jordan, M., & Abbeel, P. (2015). High-dimensional continuous control using generalized advantage estimation. arXiv preprint arXiv:1506.02438.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347.

Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., & Riedmiller, M. (2014). Deterministic policy gradient algorithms. In International conference on machine learning (pp. 387-395). PMLR.

Song, D. R., Yang, C., McGreavy, C., & Li, Z. (2018). Recurrent deterministic policy gradient method for bipedal locomotion on rough terrain challenge. In 2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV) (pp. 311-318). IEEE.

Spaan, M. T. (2012). Partially observable Markov decision processes. In Reinforcement Learning (pp. 387-414). Springer, Berlin, Heidelberg.

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

Tamar, A., Wu, Y., Thomas, G., Levine, S., & Abbeel, P. (2016). Value iteration networks. Advances in neural information processing systems, 29.

Todorov, E., Erez, T., & Tassa, Y. (2012). Mujoco: A physics engine for model-based control. In 2012 IEEE/RSJ international conference on intelligent robots and systems (pp. 5026-5033). IEEE.

Van Hasselt, H., Guez, A., & Silver, D. (2016). Deep reinforcement learning with double q-learning. In Proceedings of the AAAI conference on artificial intelligence (Vol. 30, No. 1).

Vodopivec, T., Samothrakis, S., & Ster, B. (2017). On monte carlo tree search and reinforcement learning. Journal of Artificial Intelligence Research, 60, 881-936.

Wang, C., Wang, J., Zhang, X., & Zhang, X. (2017). Autonomous navigation of UAV in large-scale unknown complex environment with deep reinforcement learning. In 2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP) (pp. 858-862). IEEE.

Wang, P., & Chan, C. Y. (2017). Formulation of deep reinforcement learning architecture toward autonomous driving for on-ramp merge. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)* (pp. 1-6). IEEE.

Wang, Z., Bapst, V., Heess, N., Mnih, V., Munos, R., Kavukcuoglu, K., & de Freitas, N. (2016). Sample efficient actor-critic with experience replay. arXiv preprint arXiv:1611.01224.

Wang, Z., Schaul, T., Hessel, M., Hasselt, H., Lanctot, M., & Freitas, N. (2016). Dueling network architectures for deep reinforcement learning.

In International conference on machine learning (pp. 1995-2003). PMLR.

Wu, Y., Mansimov, E., Grosse, R. B., Liao, S., & Ba, J. (2017). Scalable trust-region method for deep reinforcement learning using kronecker-factored approximation. Advances in neural information processing systems, 30.

Zhang, M., McCarthy, Z., Finn, C., Levine, S., & Abbeel, P. (2016a). Learning deep neural network policies with continuous memory states. In *2016 IEEE international conference on robotics and automation (ICRA)* (pp. 520-527). IEEE.

Zhang, T., Kahn, G., Levine, S., & Abbeel, P. (2016b). Learning deep control policies for autonomous aerial vehicles with mpc-guided policy search. In 2016 IEEE international conference on robotics and automation (ICRA) (pp. 528-535). IEEE.

Zhu, P., Li, X., Poupart, P., & Miao, G. (2017). On improving deep reinforcement learning for pomdps. arXiv preprint arXiv:1704.07978.

APPENDICES

APPENDIX A: Hyperparameters
APPENDIX B: Training Time
APPENDIX C: Environments and Tasks Details

APPENDIX A: Hyperparameters

The algorithms were implemented using OpenAI Spinningup. Table A.1 details the hyperparameters used in this study.

Table A.1 : Hyperparameters.

Hyperparameter	**TD3**	**SAC**	**Guided SAC**
discount factor:	0.99		
batch size:	100		
replay buffer size:	10^6		
random start step:	10000		
update after:	1000		
target NN update rate:	0.005		
optimizer:	Adam		
actor learning rate:	10^{-3}		
critic learning rate:	10^{-3}		
actor NN structure:	[256, 256]		
critic NN structure:	[256, 256]		
actor exploration noise:	0.1	-	-
target actor noise:	0.2	-	-
target actor noise clip boundary:	0.5	-	-
policy update delay:	2	-	-
entropy regulation coefficient:	-	0.2	0.2

APPENDIX B: Training Time

Table A.2 : Training Time of Each Task for 1 million Time Steps (Hour).

Task	Full Observable		Partially Observable		
	SAC	**TD3**	**SAC**	**TD3**	**Guided SAC**
HalfCheetah-v2	10.7	11	11.5	11.7	11.9
Hopper-v2	10.2	10.4	10.7	10.8	11.2
InvertedDouble Pendulum-v2	9.8	9.8	9.8	9.8	10.5
Walker2d-v2	10.6	10.8	11.6	11.6	11.8
Swimmer-v2	10	10	10.3	10.3	10.8

APPENDIX C: Environments and Tasks Details

To create partially observable environments, we modified the MuJoCo tasks listed above by deactivating the parts of the observation. A number of dimensions of the observation vector were selected and overwritten with zeros. We selected dimensions that had a significant impact on overall performance when deactivated. (Table A.3)

Table A.3 : Modification Details of the Mujoco Tasks for POMDP.

Task	Observation Space Dimensionality	Composition	Inactive Observation Dimensions
HalfCheetah-v2	17	Position: 1-8 Velocity: 9-17	5, 6, 10
Hopper-v2	11	Position: 1-5 Velocity: 6-11	4-7
InvertedDouble Pendulum-v2	11	Cart position:1 link angles sin: 2-3 link angles cos: 4-5 link velocity: 6-8 qfrc_constraint: 9-11	1, 4
Walker2d-v2	17	Position: 1-8 Velocity: 9-17	9-14
Swimmer-v2	8	Position: 1-3 Velocity: 4-8	2-6

CURRICULUM VITAE

Name Surname **:** Mehmet HAKLIDIR

EDUCATION **:**

- **B.Sc.** **:** 2003, Istanbul University, Engineering Faculty, Mechanical Engineering
- **M.Sc.** **:** 2006, Istanbul Technical University, Faculty, Mechatronics Engineering

PROFESSIONAL EXPERIENCE AND REWARDS:

- 2005 - 2022 Chief Researcher, TUBITAK BILGEM
- 2005 - … Deputy Director, TUBITAK BILGEM

PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:

- **Haklidir, M.,** & Temeltaş, H. 2021. Guided Soft Actor-Critic: A Guided Deep Reinforcement Learning Approach for Partially Observable Markov Decision Processes. *IEEE Access*, *9*, 159672-159683.

- **Haklidir, M.,** & Temeltaş, H. 2021. Derin Pekiştirmeli Öğrenme Kullanan Belirsizlik Altında Karar Verme Odaklı Otonom Sürüş Sistemleri, *30. IEEE Sinyal İşleme ve İletişim Uygulamaları Kurultayı (SIU) (pp. 1-4), 15-18 Mayıs 2022, Safranbolu, Türkiye.*

OTHER PUBLICATIONS, PRESENTATIONS AND PATENTS:

- Article (SCI Index): Gorcin Karaketir, S., Luleci, N. E., Eryurt, M.A., Emecen, A., **Haklidir, M.**, Hidiroglu, S. 2021. "Overweight and Obesity in Preschool Children in Turkey: A Multilevel Analysis", *Journal of Biosocial Science,* 1-23. doi:10.1017/S0021932022000025.

- Ozbilen M.L., Egriboz E., Halepmollasi R., Bilgen I., **Haklidir M.** 2021. "Deep Reinforcement Learning for Simulation-Based Determination of COVID-19 Pandemic Mitigation Policies". Artificial Intelligence Theory and Applications, Volume: 1, Number:3, Pages 30-39. ISSN: 2757-9778, ISBN: 978-605-69730-2-4

- Halepmollasi, R., Zeybel, M., Eyvaz, E., Arkan, R., Genc, A., Bilgen, I., **Haklidir, M.** 2021. "Towards Federated Learning in Identification of Medical Images: A Case Study", Artificial Intelligence Theory and Application, Volume: 1, Number:2 (Special Issue), Pages 30-39. ISSN: 2757-9778, ISBN: 978-605-69730-2-4

- Dervisoglu, H., Bilgen, I., Halepmollasi, R., Can, B., **Haklidir, M.** 2021. “Unfairness of Deep Learning Methods Arising Gender Bias in Covid-19 Diagnosis of Medical Images”, Artificial Intelligence Theory and Application, Volume: 1, Number:2 (Special Issue), Pages 81-94. ISSN: 2757-9778, ISBN: 978-605-69730-2-4
- Article (SCI Index): Akbayrak, I., Caglayan, S. I., Durdagi, S., Kurgan, L., Uversky, V., Ulver, B., Dervisoglu, H., **Haklidir, M.**, Hasekioglu, O., Coskuner-Weber, O. “Structures of MERS-CoV Macro Domain in Aqueous Solution with Dynamics: Impacts of Parallel Tempering Simulation Techniques and CHARMM36m and AMBER99SB Force Field Parameters”. Proteins: Structure, Function, and Bioinformatics. May 18, 2021. DOI: https://doi.org/10.1002/prot.26150
- Article (SCI Index): Haklidir Tut, F.S., **Haklidir, M**. 2020. "Prediction of geothermal originated boron contamination by deep learning approach: at Western Anatolia Geothermal Systems in Turkey", *Environmental Earth Sciences*, 79:180
- Article (SCI Index): Haklidir Tut, F.S., **Haklidir, M**. 2019. " Prediction of Reservoir Temperatures Using Hydrogeochemical Data, Western Anatolia Geothermal Systems (Turkey): A Machine Learning Approach". Natural Resources Research, 1-14.
- Article (SCI Index): Tut Haklidir, FS, **Haklidir, M.** 2017. “Fuzzy control of calcium carbonate and silica scales in geothermal systems”, Geothermics, v. 70, p. 230-238
- Article (SCI Index): **Haklidir, M.**, Tasdelen, I. 2009. “Modeling, Simulation and Fuzzy Control of an Antropomorphic Robot Arm by Using Dymola”, Journal of Intelligent Manufacturing: Volume 20, Issue2, Page 177.
- **Haklidir, M.** and Tut Haklidir, F.S. 2009. “The possibilities for hydrogen production from H2S and storage in the Black Sea”, International Journal of Nuclear Hydrogen Production and Applications, Vol. 2, No. 1, pp. 78 - 85.

Proceedings:

- Haklidir Tut, F.S., **Haklidir, M.,**” The Reservoir Temperature Prediction Using Hydrogeochemical Indicators by Machine Learning: Western Anatolia (Turkey) Case”. World Geothermal Congress 2021, Reykjavik, Iceland, May 2021.
- Ozbilen, M. L., Egriboz, E., Halepmollasi, R., Bilgen, I., **Haklidir, M.**, “A Deep Reinforcement Learning Approach to Explore Optimal Policies for Covid-19 Pandemic Mitigation: Preliminary Analysis”, II. International Artificial Intelligence in Health Congress, Izmir, Turkey, 16-18 April 2021.
- Halepmollasi, R., Zeybel, M., Eyvaz, E., Arkan, R., Genc, A., Bilgen, I., **Haklidir, M.**, “Towards Federated Learning in Identification of Medical Images: A Case Study”, II. International Artificial Intelligence in Health Congress, Izmir, Turkey, 16-18 April 2021.
- Dervisoglu, H., Bilgen, I., Halepmollasi, R., Can, B., **Haklidir, M.**, “Unfairness of Deep Learning Methods Arising Gender Bias in Covid-19 Diagnosis of Medical Images”, II. International Artificial Intelligence in Health Congress, Izmir, Turkey, 16-18 April 2021.

- Haklidir Tut, F.S., **Haklidir, M.,** "The Fluid Temperature Prediction with Hydro-geochemical Indicators Using a Deep Learning Model: A Case Study Western Anatolia (Turkey)". 43rd Workshop on Geothermal Reservoir Engineering Stanford University, CA, February 11-13-2019.
- **Haklidir, M**, Haklidir Tut, F.S., "Fuzzy Control of Calcium Carbonate and Silica Scales in Geothermal Systems", Proceedings of World Geothermal Congress, Melbourne-Australia, 19-24 April 2015.
- Aldogan, D., Senyurek, L.., **Haklidir, M.**, "Modeling and Simulation of Land Avoidance Behavior Belonging to Tactical Entities within a High Fidelity Simulation Environment", 23rd European Modeling and Simulation Symposium (EMSS 2013), Athens, Greece, September 25-27, 2013.
- Aldogan, D., **Haklidir, M.**, Eroglu, O., Franko, S., Timar, Y., Güven, A.F., Senyurek, L.., Genc, H.M., "Anti-Submarine Warfare Modeling and Simulation", 23rd European Modeling and Simulation Symposium (EMSS 2013), Athens, Greece, September 25-27, 2013.
- Ulug, U., Güven, A.F., Franko, S., **Haklidir, M.**, Sari, N., Gurbuz, M., Ozkan, U., "Case-Space Model Approach in Decision Support Systems", (In Turkish) USMOS 2013 (National Defence Simulation and Modelling Symposium) Middle East Technical University, Ankara, 11-12 June 2013.
- Aldogan, D., **Haklidir, M.**, Senyurek, L.., Timar, Y., Franko, S., Eroglu, O., Güven, A.F., Genc, H.M., "Modeling and Simulation of Anti-Submarine Warfare", (In Turkish) USMOS 2013 (National Defence Simulation and Modelling Symposium) Middle East Technical University, Ankara, 11-12 June 2013.
- **Haklidir, M.**, Franko, S., Guven, A.F., Ulug, U., "Decision Support System for Torpedo Threats (DeSuS)", Ship Survivability Conference 2010, Berlin, Germany, October 2010. – The Winner of Ship Survivability Solution Award 2010.
- **Haklidir, M.**, Tut Haklidir, F., S., "Hydrogen Production from Geothermal Sources in Turkey", World Geothermal Congress 2010 (WGC 2010), Bali, Indonesia, 25-30 April 2010.
- Aldogan, D., **Haklidir, M.**, Senyurek, L.., Koksal, S., Eroglu, O., Akdemir, C., Franko, S., Tasdelen, I., Akgun, S., "A General Approach to High Fidelity Modeling, Simulation and Control of Tactical Entities and Implementation in a Commercial Computer Generated Forces Toolkit", 21st European Modeling and Simulation Symposium (EMSS'09), Tenerife - Canary Islands, Spain, September 23-25, 2009.
- **Haklidir, M.**, Guven, A.F., Eroglu, O., Aldogan, D., Tasdelen, I., "High Fidelity Modeling and Simulation of Submarine in a Commercial Computer Generated Forces Toolkit",The Society for Modeling & Simulation International Summer Computer Simulation Conference (SCSC'09), Istanbul, Turkey, 13-16 July 2009.
- **Haklidir, M.**, Aldogan, D., Tasdelen, I., Franko, S., "Modeling, Simulation and Feedback Linearization Control of Nonlinear Surface Vessels", 6th International Conference on Informatics in Control, Automation and Robotics (ICINCO), Milan, Italy, 2-5 July 2009.

- Aruk, F., Kaplaner, A., Franko, S., Okutan, C., **Haklidir, M.**, “Full-Scale Finite Element Analysis of a Passive Sonobuoy”, Undersea Defence Technology UDT Europe 2009, Cannes, France, 9-11 June 2009.
- Franko, S., Koksal, S., **Haklidir, M.**, “Modeling, Simulation and Control Of Rotary Wing Platforms In a Computer Generated Forces Toolkit”, 7th annual Industrial Simulation Conference (ISC'2009), Loughborough, United Kingdom, June 1-3, 2009.
- Ulug, U., **Haklidir, M.**, Koruk, S., Caputcu, R., Sari, N., “Acoustic Decoy/Jammer Tactical Concepts For Submarines: A Simulation Case Study”, (In Turkish) USMOS 2009 (National Defence Simulation and Modelling Symposium) Middle East Technical University, Ankara, 17-18 June 2009.
- **Haklidir, M.**, Guven, A.F., Eroglu, O., Aldogan, D., Tasdelen, I., “Modeling and Simulation of a Submarine in a Commercial Computer Generated Forces Toolkit”, (In Turkish) USMOS 2009 (National Defence Simulation and Modelling Symposium) Middle East Technical University, Ankara, 17-18 June 2009.
- Senyurek, L., Koksal, S., Genc, H., M., Aldogan, D., **Haklidir, M.**, “Implementation of Fuzzy Control For Surface Platforms In A Computer Generated Forces Toolkit”, International Conference on Computational Intelligence for Modelling, Control and Automation (CIMCA08), Vienna, Austria, 10-12 December 2008.
- **Haklidir, M.**, Franko, S., “FST: Simulation Based Sonobouy Mechanical Design Tool”, International Sonobuoy Interoperability Conference 2008, Washington DC, USA, 23-25 September, 2008.
- **Haklidir, M.**, Aldogan, D., Tasdelen, I., “High Fidelity Modeling and Simulation of Surface Platforms in a Computer Generated Forces Toolkit”, The Society for Modeling & Simulation International [SCS] 2008 Summer Computer Simulation Conference (SCSC'08), Edinburgh, Scotland, 16-19 June 2008.
- F. Ucan, G. B. Kaplan, R. Caputcu, **M. Haklidir,** O. F. Arar, “The Solution of Tactical Shortest Path Problems Using Genetic Algorithm Approach”, (In Turkish) USMOS 2007 (National Defence Simulation and Modelling Symposium) Middle East Technical University, Ankara, 18-19 April 2007.
- **M. Haklidir**, D. Aldogan, I. Tasdelen, “Modeling And Vr-Forces Back End Integrated Simulation Of Surface Ships”, (In Turkish) USMOS 2007 (National Defence Simulation and Modelling Symposium) Middle East Technical University, Ankara, 18-19 April 2007.
- **M. Haklidir**, “FST: Simulation Based Sonobouy Mechanical Design Tool To Calculate The Replacement Of Hydrophone”, (In Turkish) USMOS 2007 (National Defence Simulation and Modelling Symposium) Middle East Technical University, Ankara, 18-19 April 2007.
- Tut Haklidir, F.S., **Haklidir, M.**, “Hydrogen Production from Geothermal Sources”, International Hydrogen Energy Congress and Exhibition IHEC 2007, Istanbul, Turkey, 13-15 July 2007.
- **Haklidir, M.**, Haklidir Tut, F., S., “Possibilities for Hydrogen Production from H2S and Storage in the Black Sea”, 5th International Conference on Hydrogen

Treatment of Material HTM 2007, Donetsk, Ukraine, May 21-25, 2007 – Key note Speaker & Co-Chairman

- **Haklidir, M.**, Tasdelen, I., "Modeling and Simulation of an Antropomorphic Robot Arm by Using Dymola", 5th International Symposium on Intelligent Manufacturing Systems, Sakarya, Turkey, 29-31 May 2006.
- **M. Haklidir**, Ş. Ertuğrul, E. Tulunay "Modeling of Sonobuoy for Position Estimation", (In Turkish) SAVTEK 2006 (National Defence Technologies Congress) Middle East Technical University, Ankara, 28-30 June 2006.
- **Haklidir, M.**, Tut, F. S., Kapkin, S., "Possibilities of Production and Storage of Hydrogen in the Black Sea", 16th World Hydrogen Energy Conference WHEC 2006, Lyon, France, 13-16 June 2006.
- **Haklidir, M.**, Kapkin, S., "Black Sea, A Hydrogen Source", International Hydrogen Energy Congress and Exhibition IHEC 2005, Istanbul, Turkey, 13-15 July 2005.
- **M. Haklidir**, M. Guler, "Fuzzy-PD Control of a Two DOF Robot Manipulator", (In Turkish) UMTS 2003 Gazi University, Ankara, 4-6 September 2003.
- M. Guler, **M. Haklidir**, N.Yagiz, "Analysis of a Planar Two-Link Robot on a Square Trajectory", Istanbul University MBGAK 2003, Istanbul, 17-20 February 2003.
- **M. Haklidir**, M. Guler, "Fuzzy Control of a One DOF Robot Arm", Istanbul University MBGAK 2003, Istanbul, 17-20 February 2003.