

# Reinforcement Learning Based Techniques in Uncertain Environments: Problems and Solutions

**Nouar AIDahoul, Zaw Zaw Htike, Rini Akmeliawati, Amir Akramin Shafie, Sheroz Khan**

*Department of Mechatronics Engineering,  
Faculty of Engineering, international Islamic University Malaysia*

## Abstract

Reinforcement learning (RL) is a well-known class of machine learning algorithms used in planning and controlling of autonomous agents. Most of the issues in planning and controlling of robots are caused by uncertainties in the actuators and sensors of robots. The paper discusses important issues faced by RL in unknown and unstructured environments. It reviews problems of RL and solutions using different variants of RL namely: hierarchical RL, Bayesian model based learning, and Partially observable Markov decision processes (POMDP).

**Keywords:** Reinforcement learning, planning, Hierarchical, Bayesian network, Partially Observable Markov decision Processes.

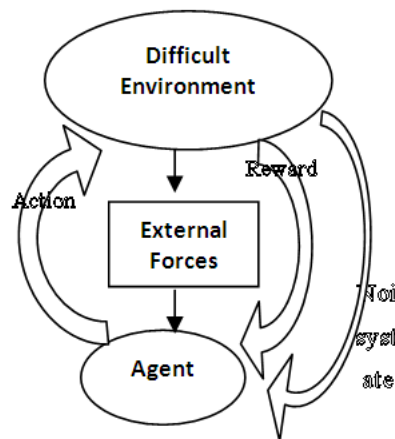
## 1. Introduction

Planning under uncertainty is important for improving the robustness of robotic systems [1]. There has been significant progress recently on robot motion planning algorithms that deal with control uncertainty, sensing uncertainty, environment changes and Model uncertainty. Agent needs to discover, by taking into consideration previously unused or uncertain actions, to get information about the punishments or rewards and the system behavior [2]. This problem is called: the exploration exploitation trade-off. Fig. 1 shows the model of agent-environment interaction in difficult conditions. In general, the environment is stochastic (non-deterministic). That means doing same action in same state may result in different next states or different reward values [3]. Above that, the state transitions probabilities or specific reward also may change with time. The change of dynamics of a robot is caused by various external factors from temperature to wear. Therefore the learning process may impossible goes to fully convergence [2]. Frequently, the environment is dynamic so

its settings cannot still the same during an earlier learning period. For example, light conditions have an impact on the vision system performance. Furthermore, the methods have to deal with uncertainties and noise or incomplete measurement and the inability to percept all states from sensors. It is also possible to have same observation with different states. This problem is called "perceptual aliasing", or "hidden state." The motion planning algorithm should also consider the collision avoidance with obstacles. The reinforcement learning algorithms should focus on these matters: small number of steps or fast learning, less of memory size (augmented capacity on line during learning), dealing with hidden states and perceptual aliasing problems, robust against noise and adaptation with uncertainties, on line learning and planning, exploration/exploitation trade off.

The paper reviews set of RL approaches that have been successfully used in complex environments. We discuss several key problems faced the robotic RL community and previous existing solutions to give reader the start point to optimize the solutions.

The paper is organized as follows: In Section 2. Various approaches in RL are discussed. In 2.1. Bayesian RL and in 2.2. Hierarchical Fig. 1 model of interaction RL, and in 2.3. different configurations of RL. In sections 2.4. and in difficult conditions 2.5. POMDP and approximation techniques are presented. Section 2.6. discusses the policy search approach. Section 3 concludes practical applications of robot in difficult environment. Finally, the paper ends with conclusion in section 4.



## 2. Different approaches in Reinforcement Learning

### 2.1. Bayesian Reinforcement Learning:

Learning with model free RL is not realistic because of cost and time or strong losses of state transition [4]. Model-based Bayesian RL can guarantee performance during and after learning. Bayesian RL represents model uncertainty as a probability Distribution function over existed models and chooses policy that maximizes long term performance. The agent begins with a prior distribution over model parameters, then after interaction with environment, posterior distribution is updated [5].

**2.1.1. exploration / exploitation trade off problem:**

- The Bayesian Exploration Exploitation Tradeoff in Learning (Beetle) frames RL as POMDPs. The optimal value functions in discrete Bayesian RL have multivariate polynomials parameters [4]. Beetle, with prior knowledge, cares about only the unknown dynamics.
- Best of Sampled Set (BOSS) uses discovery by sampling several posterior models and constructs exploring behavior that is faster in convergence to near optimality [6].

**2.1.2. Scalability (small domains) of Bayesian problem:**

- Bayes-Adaptive Monte-Carlo Planner is proper in large domains where prior knowledge makes sampling efficient[7]. It adds advances in Monte Carlo tree search to POMDP.
- a factored dynamic representation learnt by Bayesian method [8] allows powerful generalization between states with similar features. Dynamic Bayesian network benefits from the conditional independence between features to model the dynamics with few parameters.

**2.1.3. computational intractability of Bayesian problem:**

- Online Monte Carlo approach focuses on the posterior over parameters [8]. It focuses on online planning (current posterior and not all posteriors) with good computational savings.
- The BEETLE algorithm is extended to PORL in discrete domains [9]. The value function has a model whose parameters are a set of linear combinations of Dirichlets products.
- Prior Free Exploration Encouragement (PFEE) is algorithm for near Bayesian optimal policy uses a reward bonus with any prior distributions (not limited to Dirichlet) [10].

**2.1.4. Unknown and uncertain model parameters in partially observable environment**

- Bayesian particle filtering keeps the posterior distribution over models and states and finds the optimal policy. In this online planning, trajectory sampling is used to find the best policy [5]. Particle filter algorithm uses approximation of posterior as finite mixture.
- Bayesian Q-learning uses probability distributions over the Q-values [11]. The exploration and exploitation tradeoff is solved after selecting actions according to these distributions. Gaussian Process Dynamical Models (GPDMs) learn the model through the interaction with the environment. The agent works in partially observable and continuous environment without parametric form of transition, observation and reward function [12].

**2.1.5. Online Bayesian model**

Model based Bayesian RL integrates planning and learning [1]. The Bayesian POMDP is used in the continuous state space. Offline planning results in a POMDP

policy. This policy can be executed online as a finite state controller to enable the robot to learn nearly optimal policy online.

## **2.2. Hierarchical Reinforcement Learning(HRL):**

HRL decomposes the complex learning problem into small pieces to be solved easily [13]. HRL can increase the efficiency [14]. It deals with the exploration and exploitation problem. The main steps of hierarchical learning are [13]: 1) Automation of skills hierarchy design, 2) Learning of low level skills to select basic actions, 3) Learning of high level skills that coordinate basic skills. One of the hierarchical RL approaches is MAXQ decomposition. It divides the Markov decision process (MDP) target into a hierarchy of small MDPs [15]. It proves convergence with probability of 1.

### **2.2.1. scalability of HRL**

decomposition into sub-problems accelerates the learning because of these matters [15]:

- Policies that result in after learning in sub-problems can be reused and shared with others.
- The value functions that result in after learning in sub-problems can be shared.
- value function is represented as the sum of small value functions (less data for learning).
- hierarchical RL method based on action sub-rewards is used to solve convergence problem [17]. The estimate standard is whether actions are good to achieve goal or not.

### **2.2.2. Dynamic environment**

The hierarchical relative entropy policy search method [16] allows the use of hierarchical policies to learn multiple solutions at once to improve the robustness and the autonomy of the robot.

### **2.2.3. Automaton of hierarchy design**

Task hierarchy is done automatically by combining MAXQ hierarchical RL with genetic programming (GP) [18]. In this method, a subtask has a role to optimize the policy without depending on the parent task's policy. The task hierarchies resulted can help MAXQ method to learn the policy.

## **2.3. Different configurations of RL:**

### **2.3.1. Hierarchical Bayesian Reinforcement Learning**

The combination of the Hierarchical RL with Bayesian RL is very fruitful. The Hierarchical can reduce the computational complexity of Bayesian. The Bayesian gives prior knowledge about the model, policies and value functions to speed the learning and decrease the time. Hierarchical RL can decompose main goal into many sub-goals. Q Learning and Multi Layer structure [19] Benefit from Bayesian Network. planning with obstacles avoidance is automatically produced from the

learned model. The hierarchies can accelerate the learning and [19] guarantees high reliability.

### 2.3.2. Bayesian POMDP:

Sampling only possible actions that are important to compute value function is effective. Because sampling is not enough to fit large environments, an abstraction called projects parts of the transition dynamics are used. It focuses on the promising areas [20].

### 2.3.3. hierarchical POMDP

Hierarchical approach can solve complex POMDPs that are intractable. The Robot Navigation Hierarchical POMDP (RN-HPOMDP) structure divides a flat POMDP with large action and state spaces into several POMDPs [21]. Therefore, in the upper levels level, there is a coarse discretization of states and actions. These are called "abstract states and actions".

### 2.3.4. Bayesian hierarchical POMDP:

Bayesian hierarchical RL is formulated as a partially observable semi-MDP (POSMDP) [22]. It samples from a prior belief to build an approximate model for each POSMDP, then solve using Monte Carlo Value Iteration with Macro-Actions solver.

## 2.4. Partially observable Markov decision processes techniques

Planning has two problems [23]. "curse of dimensionality": high dimensional belief, and "curse of history" or "long time horizon". In a motion planning mission, an agent always needs to take large number of actions. Uncertainty can guide the robot to perform untrue actions [21]. POMDPs model the hidden states that are partially observable and keep a belief distribution over the states. A POMDP uses both a priori model with the history of observations and actions to find the belief [24].

### 2.4.1. Memory size problem

- Variable-Resolution Percept Discretization works effectively in noisy and continuous worlds to differ perceptually aliased states [25]. It dynamically changes the discretization to guarantee generalization: fine when reward changes quickly, and coarse when the smooth reward.

### 2.4.2. Model parameters of POMDP:

The problem of the Bayesian POMDP algorithm has computational cost and depends on environment models [26]. getting exact parameters makes it complex mission because of large data required.

- A particle filter algorithm has posterior distribution and plans online by sampling trajectory [5]. Updating of posterior distribution can be done online after getting new observation.

- Hierarchical POMDPs for navigation of autonomous agent can efficiently model large environments at a fine resolution and employs hierarchy of state and action spaces [21].

## 2.5. POMDP approximation techniques

The intractability comes from large computation of an exact optimal policy for the whole belief space [24]. The POMDP approximation methods can only solve problems with limited size of state space. The tradeoff between the approximations and efficiency is an important topic.

### 2.5.1. reduction of belief space problem

Belief space must be small enough to overcome the computation difficulty and large enough to get good approximation. POMDP value iteration focuses on how to collect belief state points.

- Point-Based Value Iteration (PBVI) for planning approximates VI by choosing a limited number of representative belief points. It uses stochastic trajectories to select points and keeps one hyper-plane per point. It updates both value and gradient to generalize better [27].
- PBVI randomized approximate VI [28] plans on randomly sampled points of reachable belief. It deals with highly perceptual aliases. As a result of back up, the value of all points improved.

### 2.5.2. Online Planning / execution problem

It is impossible for any offline point-based value iteration algorithm to traverse the entire reachable belief states space in the limited time (the history curse problem).

- Online planning approaches are used to solve the historical curse problem effectively [29]. Regarding to time constraint, online algorithm has two phases: planning phase and execution phase. It builds an ideal POMDP model that meets the real time system requirements.
- Adaptive Belief Tree (ABT) doesn't replan from scratch. but reuse and improve existing solution [30], and update the solution whenever the POMDP model changes during runtime.
- Point based online value iteration (PBOVI) algorithm keeps the quality of offline policies and meets the requirements of real time [29]. It uses branch-and-bound pruning algorithm to the AND/OR tree of belief states online. Previously belief states are reused.

### 2.5.3. POMDP curse of history problem:

In motion planning mission, an agent often takes a large number of actions to arrive to its destination. Therefore, the complexity grows exponentially with the time.

- Point based POMDP solver, called Milestone Guided Sampling [23] works by sampling a group of points, called milestones, from an agent's state space. It uses the representation to guide sampling in the belief space.
- Partially Observable Monte Carlo Planning (POMCP) approach combines a Monte Carlo updating of belief with a Monte Carlo tree searching from the

current belief [31]. It begins by sampling states from the belief state, after that, sampling histories using a black box simulator. It is useful in large problems that is difficult to be represented by explicit distributions.

#### 2.5.4. Belief state dimensionality problem:

To keep trade-off between not losing of important details and reducing dimensions as possible:

- Exponential family Principal Components Analysis (E-PCA) is used for large POMDPs [24] and reduces the belief space dimensionality by exploiting few features from the belief state. its reduction guarantees no loss of information and the reconstruction by keeping same variance. it falls in the policy computation because the nonlinearity (non POMDP) and that destroys the PWLC property of the value function.
- (AMDP) uses entropy and maximum likelihood state to present belief states [32]. It generalizes models with continuous states. Parametric POMDP reduces the dimensions by presenting belief as low dimensions of multimodal or heavy tail parameterized distributions.

#### 2.5.5. Finite state controller problem:

FSC is a way to represent policies for POMDPs.

- Bounded policy iteration approach [33] constructs good controller that combines gradient ascent (efficiency and searching in bounded space) and policy iteration (less sensitive to local optima).
- GA-FSC algorithm which is an approximate of DEC-POMDP uses finite state controllers (FSC) to represent a finite-horizon policy [34]. It searches the policy space when either the size or horizon is large. an approximate fitness function is used to reduce computation time.

#### 2.5.6. full backup POMDP problem:

Some backups can be removed without having impact

- A randomized point-based value iteration (Perseus) uses single backup to improve the value of several belief points (the belief space is improved in each backup stage).
- The Prioritized Value Iteration (PVI) algorithm [35] orders backup just on a pre known group of belief points then prioritizes backups in the POMDP. Prioritizing the order of identical backups on states is necessary to converge quickly.

#### 2.6. Policy Search as an alternative to Value Function:

value function approximation leads to high bias and convergence problem [36]. A little change in the policy can lead to big change in the value function which in its turn leads to big change in the policy.

- RL gradient-based constructs a class of parameterized policies [36]. It is scalable to control POMDPs in memory-less problems and learn how to do action and what to remember [37].

- Policy search takes the current policy and its neighborhood to enhance the performance. It finds good parameters for a policy by maximizing the expected reward [38].
- Policy gradient RL solves continuous states problems [39] and guarantees the convergence to local optimal policy (gradient large variance). It uses prior knowledge of policy parameters.
- hierarchical policy gradient [39] can incorporate prior knowledge and decompose the task into a set of subtasks with smaller states. It scales PGRL to high dimensional domains.
- Policy Evaluation-of-Goodness And Search Using Scenarios can reduce the variance. It searches in policies space for MDP or POMDP given a deterministic simulative model [40].

### **3. Examples of robots working in difficult environment**

#### **3.1. AUV:**

Autonomous Underwater Vehicle has complex nonlinear dynamics [19]. The fluid dynamical forces and moments cannot be predicted easily because of their sophisticated characteristics. Above that, the environment has unknown obstacles like rocks and ship wrecks [19]. Moreover, the speed of water current often exceeds the speed of AUV. AUV which is equipped with imaging sensor is used to classify objects. The classification is modeled as POMDP [41]. The objective is to focus on dynamically the next aspect to classify more quickly. Another AUV looks for hydrothermal vents that are existing on the sea floor with noisy information of temperature and chemical sensors [42].

#### **3.2. UAV (Unmanned Aerial Vehicle) : [43]**

When the uncertainties come from imperfect action or observation, policies can be automatically optimized by POMDP. The decision, targets number, zones and positions of targets, are usually unknown before the flight. This problem is considered as a long-term sequential decision planning with both perception actions (view angle of camera) and mission actions (moving, landing). POMDP is solved online during the flight. Collision avoidance is modeled by POMDP [44]. Monte Carlo Value Iteration allows POMDP models with continuous state to output a policy graph.

#### **3.3. armed robot:**

In the task of two-armed robot replacing a flat tire [45], objects localization is based on noisy sensors. Above that, it may be impossible to determine if some subtasks were executed correctly. Therefore, the robot and its environment are modeled as a POMDP. Another example is a robot arm that equipped with pair of stereo camera which gives noisy observations of end effectors position [46]. Another application is palletizing robot [47]. There is large number of uncertainties for the path planning of multiple joints robot. the Hierarchical MDP model reduces states and searches faster.

#### 4. Conclusions

From this survey, it can be deduced that using RL techniques in the field of robotics is not an easy direct task but rather requires number of steps before getting the optimal solution. The paper discusses important aspects of RL in unknown and unstructured environment including Hierarchical RL, Bayesian model, and Partially Observable Markov decision Process theory.

#### References

- [1] Haoyu Bai, D. H. (2013). Planning How to Learn. *2013 IEEE International Conference on Robotics and Automation (ICRA)*.
- [2] Kober, J., & Peters, J. (2012). Reinforcement Learning in Robotics A Survey. *The International Journal of Robotics Research*.
- [3] Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement Learning: A Survey. *Journal of Artificial Intelligence Research*.
- [4] Poupart, P., Vlassis, N., Hoey, J., & Regan, K. (2006). An Analytic Solution to Discrete Bayesian Reinforcement Learning. *Proceedings of the 23 rd International Conference on Machine Learning*.
- [5] Dallaire, P., Ross, S., & Chaib-draa, B. (2009). GP-POMDP: Bayesian Reinforcement Learning in Continuous POMDPs with Gaussian Processes. *Intelligent Robots and Systems, IROS IEEE/RSJ International Conference on*.
- [6] Asmuth, J., Li, L., Littman, M. L., Nouri, A., & Wingate, D. (2009). A Bayesian Sampling Approach to Exploration in Reinforcement Learning. *UAI '09 Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*.
- [7] Guez, A., Silver, D., & Dayan, P. (2013). Scalable and Efficient Bayes-Adaptive Reinforcement Learning Based on Monte-Carlo Tree Search. *Journal of Artificial Intelligence Research*.
- [8] Ross, S., & Pineau, J. (2008). Model-Based Bayesian Reinforcement Learning in Large Structured Domains. *Proceedings of the Twenty-Fourth Conference on Uncertainty in Artificial Intelligence (UAI2008)*.
- [9] Poupart, P., & Vlassis, N. (2008). Model-based Bayesian Reinforcement Learning in Partially Observable Domains. *2008 ISAIM (International Symposium on Artificial Intelligence and Mathematics)*.
- [10] Kawaguchi, K., & Sato, H. (2013). Prior-free exploration bonus for and beyond near bayes optimal behavior. *IJCAI '13 Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*.
- [11] Dearden, R., Friedman, N., & Russell, S. (1998). Bayesian Q-learning. *AAAI/Innovative Applications of Artificial Intelligence Conference*.

- [12] Dallaire, P., Ross, S., & Chaib-draa, B. (2009). GP-POMDP: Bayesian Reinforcement Learning in Continuous POMDPs with Gaussian Processes. *Intelligent Robots and Systems, IROS IEEE/RSJ International Conference on.*
- [13] Lin, L.-J. (1993). *Reinforcement Learning for Robots Using Neural Networks*. PhD thesis, Carnegie Mellon University, School of Computer Science.
- [14] Mahajan, S. (2014). Hierarchical Reinforcement Learning in Complex Learning Problems: A Survey. *International Journal of Computer Science and Engineering.*
- [15] Thomas G. Dietterich. (2000). Hierarchical Reinforcement Learning with the MAXQ Value Function Decomposition. *Journal of Artificial Intelligence Research.*
- [16] Daniel, C., Neumann, G., & Peters, J. (2013). Autonomous Reinforcement Learning with Hierarchical REPS. *Neural Networks (IJCNN), The 2013 International Joint Conference on.*
- [17] Fu, Y., Liu, Q., Ling, X., & Cui, Z. (2014). A Reward Optimization Method Based on Action Subrewards in Hierarchical Reinforcement Learning. *The Scientific World Journal.*
- [18] Elfwing, S., Uchibe, E., Doya, K., & Christensen, H. I. (2007). Evolutionary Development of Hierarchical Learning Structures. *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION.*
- [19] KAWANO, H., & URA, T. (2002). Motion Planning Algorithm for Non-Holonomic Autonomous Underwater Vehicle in Disturbance using Reinforcement Learning and Teaching Method. *Proceedings of the 2002 IEEE International Conference on Robotics and Automation.*
- [20] Acuna, D., & Schrater, P. (2009). Improving Bayesian Reinforcement Learning Using Transition Abstraction. *Proceedings of the ICML/UAI/COLT Workshop on Abstraction in Reinforcement Learning.*
- [21] Foka, A., & Trahanias, P. (2005). Real-Time Hierarchical POMDPs for Autonomous Robot Navigation. *Robotics and Autonomous Systems.*
- [22] Vien, N. A., Ngo, H., & Ertel, W. (2014). Monte Carlo Bayesian Hierarchical Reinforcement Learning. *Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2014).*
- [23] Kurniawati, H., Du, Y., Hsu, D., & Lee, W. S. (2010). Motion Planning under Uncertainty for Robotic Tasks with Long Time Horizons. *International Journal of Robotics Research.*
- [24] Roy, N., & Thrun, S. (2005). Finding Approximate POMDP Solutions Through Belief Compression. *Journal of Artificial Intelligence Research.*

- [25] Broadbent, R., & Peterson, T. (2005). Robot Learning in Partially Observable, Noisy, Continuous Worlds. *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*.
- [26] Sutton, R. S., & Barto, A. G. (1998). *Reinforcement Learning An Introduction*. A Bradford Book The MIT Press.
- [27] Pineau, J., Gordon, G., & Thrun, S. (2003). Point-based value iteration: An anytime algorithm for POMDPs. *IJCAI*.
- [28] Spaan, M. (2004). A point-based POMDP algorithm for robot planning. *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*.
- [29] BoWu, Zheng, H.-Y., & Feng, Y.-P. (2014). Point-based online value iteration algorithm in large POMDP. *Journal of Applied Intelligence, Springer*.
- [30] Kurniawati, H., & Yadav, V. (2013). An Online POMDP Solver for Uncertainty Planning in Dynamic Environment. *International Symposium on Robotics Research*.
- [31] Silver, D., & Veness, J. (2010). Monte-Carlo Planning in Large POMDPs. *Advances in Neural Information Processing System NIPS*.
- [32] Zhou, E., Fu, M. C., & Marcus, S. I. (2010). Solving Continuous-State POMDPs via Density Projection. *IEEE Automatic Control, IEEE Transactions on*.
- [33] Poupart, P., & Boutilier, C. (2003). Bounded Finite State Controllers. *Advances in neural information processing systems*.
- [34] Eker, B., & Akin, H. L. (2013). Solving decentralized POMDP problems using genetic algorithms. *Autonomous Agents and Multi-Agent Systems*.
- [35] Shani, G., Brafman, R. I., & Shimony, S. E. (2008). Prioritizing Point-Based POMDP Solvers. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*.
- [36] Baxter, J., & Bartlett, P. L. (2000). Reinforcement Learning in POMDP's via Direct Gradient Ascent. *In Proc. 17th International Conf. on Machine Learning*.
- [37] D.A., A. (2003). *Policy Gradient Algorithms for Partially Observable Markov Decision Processes*. PhD Thesis, Australian National University.
- [38] Deisenroth, M. P., Neumann, G., & Peters, J. (2013). A Survey on Policy Search for Robotics. *Foundations and Trends in Robotics*.
- [39] Ghavamzadeh, M., & Mahadevan, S. (2003). Hierarchical Policy Gradient Algorithms. *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*.

- [40] Andrew Y. Ng, & Jordan, M. (2000). PEGASUS: A policy search method for large MDPs and POMDPs. *UNCERTAINTY IN ARTIFICIAL INTELLIGENCE PROCEEDINGS*.
- [41] Myers, V., Defence R&D Canada - Atlantic, D. N., & Williams, D. (2011). Adaptive Multiview Target Classification in Synthetic Aperture Sonar Images Using a Partially Observable Markov Decision Process. *Oceanic Engineering, IEEE Journal of (Volume:37, Issue: 1)*.
- [42] Dearden, R., Saigol, Z. A., Wyatt, J. L., & Murton, B. J. (2007). Planning for AUVs: Dealing with a Continuous Partially-Observable Environment. *17th International Conference on Automated Planning & Scheduling*.
- [43] P, C., Chanel, C., Teichteil-Konigsbuch, F., & Lesire, C. (2012). POMDP-based online target detection and recognition for autonomous UAVs. *ECAI conference*.
- [44] Bai, H., Hsu, D., Kochenderfer, M. J., & Lee, W. S. (2011). Unmanned Aircraft Collision Avoidance using Continuous-State POMDPs. *Robotics: Science and Systems*.
- [45] Burdick, Sharan, R., & Joel. (2014). Finite State Control of POMDPs with LTL Specifications. *American Control Conference (ACC)*.
- [46] Berg, J. v., Abbeel, P., & Goldberg, K. (2011). LQG-MP: Optimized path planning for robots with motion uncertainty and imperfect state information. *The International Journal of Robotics Research*.
- [47] Liu, J., Wang, Z., Chen, Z., Yang, Z., Wang, Z., & Liu, C. (2014). Hierarchical Markov Decision Based Path Planning for Palletizing Robot. *TELKOMNIKA Indonesian Journal of Electrical Engineering*.