ISSN: 2454-9940

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

E-Mail :
editor.ijasem@gmail.com
editor@ijasem.org

www.ijasem.org

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN:2454-9940

www.ijsem.org

Vol 14, Issue.3 july 2023

COMPARISON OF DEEP REINFORCEMENT LEARNING AND MODEL PREDICTIVE CONTROL FOR ADAPTIVE CRUISE CONTROL

[1]I ROHINI,[2]PALLIKONDA APOORVA, [3]Y.SAI BHAVANI, [4]CH.YAMUNA,[5]B. JYOTHI

ABSTRACT:This study compares Deep Reinforcement Learning (DRL) and Model Predictive Control (MPC) for Adaptive Cruise Control (ACC) design in car-following scenarios. A first-order system is used as the Control-Oriented Model (COM) to approximate the acceleration command dynamics of a vehicle. Based on the equations of the control system and the multi-objective cost function, we train a DRL policy using Deep Deterministic Policy Gradient (DDPG) and solve the MPC problem via InteriorPoint Optimization (IPO). Simulation results for the episode costs show that, when there are no modeling errors and the testing inputs are within the training data range, the DRL solution is equivalent to MPC with a sufficiently long prediction horizon. Particularly, the DRL episode cost is only 5.8% higher than the benchmark solution provided by optimizing the entire episode via IPO. The DRL control performance degrades when the testing inputs are outside the training data range, indicating inadequate generalization. When there are modeling errors due to control delays, disturbances, and/or testing with a High-Fidelity Model (HFM) of the vehicle, the DRL-trained policy performs better with large modeling errors while having similar performance as MPC when the modeling errors are small.

Index Terms—Deep Reinforcement Learning, Model Predictive Control, Adaptive Cruise Control.

INTRODUCTION

Reinforcement learning is a learning-based method for optimal decision making and control [1]. In reinforcement learning, an agent takes an action based on the environment state and consequently receives a reward. Reinforcement learning maximizes cumulative discounted reward by learning an optimal state-action mapping policy through trial and error. The policy is trained via Bellman's principle of optimality, which dictates that the remaining actions constitute an optimal policy with regard to the state resulting from a previous action. Deep reinforcement learning (DRL), which utilizes deep (multi-layer) neural netsas

[1]ASSISTANT PROFESSOR, DEPARTMENT OF EEE, MALLA REDDY ENGINEERING COLLEGE FOR WOMEN, HYDERABAD

[2,3,4&5]UGSCHOLAR, DEPARTMENT OF IOT, MALLA REDDY ENGINEERING COLLEGE FOR WOMEN, HYDERABAD

policy representations, has drawn significant attention as its trained policy surpassed the best human in playing board games [2]. Different DRL algorithms have been proposed which include Deep Q-Networks [3], Trust Region Policy Optimization [4], Proximal Policy Optimization [5], and Deep Deterministic Policy Gradient (DDPG) [6]. In this work, we use DDPG, which outputs continuous control actions by training a deterministic policy offline. DDPG is a popular choice for optimal control, especially for a stable dynamic system [7]. Model Predictive Control (MPC) represents the state of the art for the practice of real-time optimal control [8]. MPC benefits from a sufficiently accurate model of the plant dynamics. At each time step, a constrained optimization problem is formulated based on the plant model to minimize a defined cost function in a predictive time horizon. The optimization problem is solved online and only the first value of the solved control sequence is applied. At the next time step, this predictive control procedure is repeated with updated states. There are various methods to formulate the optimization problem with the state-space equations and the cost function, which include direct single shooting, direct multiple shooting, and direct collocation [9]. There are also various online optimization solvers for MPC, which include sequential quadratic programming and IPO [8]. In this work, we use IPO with direct single shooting, which solves the formulated optimization problem via Newton-Raphson's method by successively approximating the root of the cost function derivative [10]. The IPO solution is on the interior of the set described by the inequality constraints and close to the true optimal solution. Since both DRL and MPC can provide optimal control solutions, it is of research interest to understand their advantages and disadvantages. For our comparison, we consider solving an optimal control problem for a dynamic system represented by a system of state-space equations. We do not consider training an end-to-end (such as image-to-control-action) solution using DRL [3]. Before using an example for comparison, one could understand some known differences between the two. Firstly, MPC demands online optimization that requires relatively powerful computing devices for real-time applications, which raises monetary concerns. For automotive engineering, hardware-in-the-loop simulations are needed to verify the real-time readiness of MPC before real-world deployment [11].

On the other hand, offline-trained DRL solutions are neural nets that result in very little computation time during deployment. Secondly, MPC is model-based while, up to date, DRL control solutions are black-box neural nets that lack theoretical assurance [12]. In this work, we do not focus on these known differences about the computing requirements and theoretical assurance for DRL and MPC. In this work, we focus on the optimality level (minimum episode cost) that DRL and MPC can achieve without and with modeling errors. For fair comparison, we use the same COM of the vehicle for DRL to train a policy and for MPC optimization. Most of the parameter settings are the same for both DRL and MPC except that the DRL reward utilizes a discount factor that is absent in the MPC optimization. This is due to the fact that DRL usually requires a discount factor less than one for convergence [13] while MPC normally does not include the discount factor. We raised a few questions that guided our research: (1) When there are no modeling errors, for example, testing on the vehicle COM, is DRL or MPC better in achieving the minimum cost? We use IPO to optimize for the entire simulation episode once to obtain a benchmark solution, called the IPO solution, for both DRL and MPC. Note that the IPO solution is not a receding-horizon one since it's obtained by setting the predictive time horizon as the episode length and the optimization is solved only once. MPC usually obtains better optimality levels with longer prediction horizons. It may be interesting to see the difference between the DRL solution and MPC with different prediction horizons. The comparison of the DRL, MPC, and IPO solutions could provide insights on training policies via Bellman's principle of optimality versus optimizing via Newton-Raphon's method. It would also show the effect of the discount factor on the optimality-seeking of DRL. Additionally, we also want to investigate if the machine learning generalization issue persists in the DRL-trained neural net. When the testing inputs are outside the range of training data, the DRL control performance may be compromised and lose competitiveness to MPC. (2) When there are modeling errors, does DRL or MPC achieve a lower cost? Modeling errors in this paper refer to the differences from the ACC car-following state-space equations. Such modeling errors include neglected control delays, disturbances, and/or the difference between the COM and HFM. In our previous work, we showed that modeling errors due to neglecting vehicle dynamics could cause significantly degraded DRL control performance [14]. As both DRL and MPC suffer from performance degradation due to modeling errors [15], [16], this work could show whether DRL or MPC is better at handling modeling errors given that most conditions are the same. It's worth mentioning that DRL has been shown to perform better than a rule-based method for lane-change control in the presence of environment noise [17]. To answer the raised questions, we develop both DRL and MPC controllers for ACC car-following control. Car following is one of the most common behaviors of road vehicles [18]. ACC is a type of Advanced Driver Assistance System that enables intelligent and automated driving [19]. Automated vehicle development has been a popular interest in academia and industry as it could potentially revolutionize transportation. We develop ACC controllers for a power-split plug-in hybrid electric vehicle (PHEV), a 2015 Toyota Prius, since we have previously developed a HFM of it in MATLAB/Simulink [11], [20]. The HFM includes control input execution delay (control delay) of 0.2s, powertrain modeling, and external resistances including aerodynamic drag and rolling resistance. Road grade is not considered as we assume flat surfaces. The complexity of the HFM can be shown by its powertrain modeling, see Fig. 1. The powertrain modeling of the HFM includes the modeling of its battery, battery converter, electric motors, combustion engine, and planetary gears. In addition, the HFM includes a rule-based energy management system (charge-depletioncharge-sustaining) to determine the power demands for the battery and engine [21]. The HFM is based on Autonomie, a MATLAB/Simulink simulation tool for automotive control developed by the Argonne National Lab. Note that the firstorder vehicle COM considered in this work does not include the control delay.

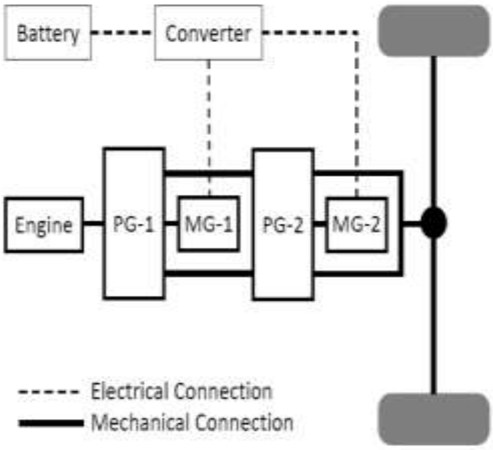

Fig. 1. Schematic of the HFM powertrain of a 2015 Toyota Prius power-split PHEV. In the plot, PG means planetary gear and MG means motor-generator.

We acknowledge that our comparison of MPC and DRL is limited to a certain scope when considering the effect of modeling errors. On one hand, there are more advanced MPC and DRL methodologies. Our adopted MPC methodology that includes direct single shooting and IPO is typical yet simple. Research advances in tube-based and stochastic MPC could make MPC more robust and disturbance-tolerant [15], [16], [22]. Regarding DRL, our adopted DDPG algorithm is a cornerstone but could be polished. The use of transfer learning and/or meta-learning on the DRL-trained policy could make it better in handling modeling errors and uncertainties [23], [24]. On the other hand, the ACC car-following control example is a low-dimensional task with only three state variables while DRL is known to handle well higher-dimensional tasks with complex cost functions [6], [25]. For the scope of this work, we only consider the low-dimensional task without considering robust and stochastic MPC or transfer and meta-learning. The main contribution of this work is the quantitative and comprehensive comparison of the well-known DRL algorithm, DDPG, and an MPC that is based on the popular IPO method. We consider the effect of the MPC prediction horizon, the generalization issue of DRL, the case of no modeling errors, and the cases of modeling errors that include the control delay, disturbances, and testing with the HFM. To our best knowledge, there is no such comparison existing in the literature. We hope that such a comprehensive comparison will serve as a useful reference for researchers working on optimal control.

LITERATURE REVIEW There are only a limited number of papers in the literature that compare reinforcement learning and MPC performances. In [26], the authors compared reinforcement learning and MPC in controlling non-linear electrical power oscillation damping. With a random tree as the policy, the reinforcement learning is not DRL. With a low-dimensional deterministic model of the system, the authors considered no modeling errors. The results show that with different parameter settings, the reinforcement learning solutions could be worse or better than MPCs with regard to the cumulative discounted cost. The authors also showed data that indicates that reinforcement learning is at least 10 times faster than MPC during testing. In [27], the authors compared DRL and receding-horizon control (same as MPC) in controlling a team of unmanned aerial vehicles to maximize wild fire coverage.

The authors used a stochastic model of wild fire propagation that adds randomness (disturbances) to the control. The DRL environment state is high-dimensional since it includes both images and continuous states, indicating a hybrid-input DRL control. The results show that DRL outperformed receding-horizon control by a moderate margin regarding cumulative reward. In [25], the authors compared integrated MPC-DRL and pure MPC controllers for control of high-dimensional tasks such as 3D humanoid standing up from the ground and in-hand manipulation by a five-fingered robotic hand. The integrated MPC-DRL controller is essentially a MPC controller wherein the MPC terminal cost is learned via DRL. The training and testing were based on an accurate model without considering modeling errors. The authors found that the integrated MPCDRL controller achieved higher rewards than a pure MPC controller by a moderate margin. In [28], the authors compared DRL and MPC for merging into dense traffic. The DRL and MPC methods do not share the same cost function. Specifically, DRL has a complex cost function including absolute-value and linear costs while MPC has a quadratic cost function. Thus, the authors did not compare the episode costs of DRL and MPC. However, the authors found that the DRL-trained policy significantly outperformed MPC regarding the rate of merging success. In summary, there is a lack of literature on comparing DRL and MPC in a fair manner, especially in the presence of modeling errors. Our motive originates from solving a traditional optimal control problem that can be represented by state-space equations. In our work, most conditions are set to be the same for DRL and MPC for fair comparison. The HFM of Prius enables us to study the effect of practicallyexisting modeling errors on the control performances of DRL and MPC. These characteristics make our work different from the existing literature. There is also limited literature on ACC car-following control using DRL. In [29], the authors used a single-layer (non-deep) neural net as the reinforcement learning policy representation to train an ACC controller. In [30], [31], naturalistic driving data was used to train human-like car-following policies using DRL. In our previous work, we trained an ACC optimal control policy with a state-space car-following model using DRL for the first time [14]. Our previous work is the base for the DRL controller development in this paper. However, the car-following model in this paper considers a constant time headway instead of a constant distance headway in the previous work. The constant time headway enables the vehicle to proportionally adjust the desired inter-vehicular distance based on its speed, which is more appropriate in real-world driving. There is a large body of literature on ACC using MPC [11], [15], [32], [33]. In such research papers, the modelpredictive ACC systems were designed with multi-objective cost functions to minimize the tracking error, energy consumption, vehicle jerk, and etc. A first-order system was usually considered to be sufficient to approximate the acceleration command dynamics of the vehicle [34], [35]. The first-order approximation is due to the imperfect estimation of vehicle parameters, lower-level control of acceleration and brake pedals' positions, unmodeled powertrain dynamics, and external disturbances [33]. Our ACC problem formulation described in the following section is similar to that from the modelpredictive ACC papers

EXISTING SYSTEM

In existing system, Support Vector Machines (SVM), decision tree classifier, random forest regression, and neural network [10]. Even though there are many algorithms to choose, only specific algorithms are suitable to make certain predictions. In this paper, a machine learning algorithm is applied to predict a met material Thermal parameters,

DISADVANTAGES

- Doesn't Efficient for handling large volume of data.
- Theoretical Limits
- Incorrect Classification Results.
- Less Prediction Accuracy.

PROPOSED SYSTEM

The proposed model is introduced to overcome all the disadvantages that arises in the existing system. This system will increase the accuracy of the classification results by classifying the data based on the Smart Grid prediction dataset and others using LSTM algorithms.It enhances the performance of the overall classification results.

ADVANTAGES

- High performance.
- Provide accurate prediction results.
- It avoid sparsity problems.
- Reduces the information Loss and the bias of the inference due to the multiple estimates.

IMPLEMENTATION

MODULES

- Data Selection and Loading
- Data Preprocessing
- Splitting Dataset into Train and Test Data
- Classification
- Prediction
- Result Generation

MODULES DESCRIPTION

DATA SELECTION AND LOADING

- Data selection is the process of determining the appropriate data type and source, as well as suitable instruments to collect data.
- Data selection precedes the actual practice of data collection and it is the process where data relevant to the analysis is decided and retrieved from the data collection.
- In this project, the Smart Grid dataset

DATA PREPROCESSING

- The data can have many irrelevant and missing parts. To handle this part, data cleaning is done. It involves handling of missing data, noisy data etc.
- Missing Data:
  This situation arises when some data is missing in the data. It can be handled in various ways.
  - ✓ Ignore the tuples: This approach is suitable only when the dataset we have is quite large and multiple values are missing within a tuple.
  - ✓ Fill the Missing values:
    There are various ways to do this task. You can choose to fill the missing values manually, by attribute mean or the most probable value.
- Encoding Categorical data: That categorical data is defined as variables with a finite set of label values. That most machine learning algorithms require numerical input and output variables. That an integer and one hot encoding is used to convert categorical data to integer data.

- Count Vectorizer: Scikit-learn's CountVectorizer is used to convert a collection of text documents to a vector of term/token counts. It also enables the pre-processing of text data prior to generating the vector representation. This functionality makes it a highly flexible feature representation module for text.

SPLITTING DATASET INTO TRAIN AND TEST DATA

- Data splitting is the act of partitioning available data into two portions, usually for cross-validator purposes.
- One Portion of the data is used to develop a predictive model and the other to evaluate the model's performance.
- Separating data into training and testing sets is an important part of evaluating data mining models.
- Typically, when you separate a data set into a training set and testing set, most of the data is used for training, and a smaller portion of the data is used for testing.
- To train any machine learning model irrespective what type of dataset is being used you have to split the dataset into training data and testing data.

CLASSIFICATION

Classification is the problem of identifying to which of a set of categories, a new observation belongs to, on the basis of a training set of data containing observations and whose categories membership is known.

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean/average prediction (regression) of the individual trees.

Decision Trees are a type of Supervised Machine Learning (that is you explain what the input is and what the corresponding output is in the training data) where the data is continuously split according to a certain parameter. An example of a decision tree can be explained using above binary tree.

The SVM is one of the most powerful methods in machine learning algorithms. It can find a balance between model complexity and classification ability given limited sample information. Compared to other machine learning methods, the SVM has many advantages in that it can overcome the effects of noise and work without any prior knowledge. The SVM is a non-probabilistic binary linear classifier that predicts an input to one of two classes for each given input. It optimizes the linear analysis and classification of hyperplane formation techniques.

The NN algorithm is mainly used for classification and regression in machine learning. To determine the category of an unknown sample, all training samples are used as representative points, the distances between the unknown sample and all training sample points are calculated, and the NN is used. The category is the sole basis for determining the unknown sample category. Because the NN algorithm is particularly sensitive to noise data, the K-nearest neighbour algorithm (KNN) is introduced. The main concept of the KNN is that when the data and tags in the training set are known, the test data are input, the characteristics of the test data are compared with the features corresponding to the training set, and the most similar K in the training set is found.

PREDICTION

Predictive analytics algorithms try to achieve the lowest error possible by either using “boosting” or “bagging”.

Accuracy − Accuracy of classifier refers to the ability of classifier. It predict the class label correctly and the accuracy of the predictor refers to how well a given predictor can guess the value of predicted attribute for a new data.

Speed − Refers to the computational cost in generating and using the classifier or predictor.

Robustness − It refers to the ability of classifier or predictor to make correct predictions from given noisy data.

Scalability − Scalability refers to the ability to construct the classifier or predictor efficiently; given large amount of data.

Interpretability − It refers to what extent the classifier or predictor understands.

RESULT GENERATION

The Final Result will get generated based on the overall classification and prediction. The performance of this proposed approach is evaluated using some measures like,

- Accuracy

Accuracy of classifier refers to the ability of classifier. It predicts the class label correctly and the accuracy of the predictor refers to how well a given predictor can guess the value of predicted attribute for a new data.

A

C= $\frac{TP+TN}{TP+TN+FP+FN}$

- Precision

Precision is defined as the number of true positives divided by the number of true positives plus the number of false positives.

Pre

cision= $\frac{TP}{TP+FP}$

- Recall

Recall is the number of correct results divided by the number of results that should have been returned. In binary classification, recall is called sensitivity. It can be viewed as the probability that a relevant document is retrieved by the query.

- ROC

ROC curves are frequently used to show in a graphical way the connection/trade-off between clinical sensitivity and specificity for every possible cut-off for a test or a combination of tests. In addition the area under the ROC curve gives an idea about the benefit of using the test(s) in question.

- Confusion matrix

A

confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known. The confusion matrix itself is relatively simple to understand, but the related terminology can be confusing.

CONCLUSION AND FUTURE WORK

The aim of this paper was to evaluate different control strategies for thermal energy management in buildings. Three cutting-edge solutions, model predictive control and deep reinforcement learning with offline and online 806 training were tested and analyzed on a simple case study system and bench- 807 marked against a classical rule-based control approach. The objective of the controllers was to satisfy the cooling demand of a small office building while minimizing the cost of electricity drawn from the grid to operate the chiller, 810 making the best use of a thermal energy storage tank. The controllers could manage the amount of energy charged and discharged to/from a cold water storage by adjusting the water mass flow rate circulated to it. MPC is a model-based solution that employs a simplified model of the controlled system to perform an optimization process over a receding horizon,

using predictions of external disturbances. Similarly, DRL employs predic- tions of external disturbances to learn a near-optimal control policy. However, despite the model-free nature of the control algorithm, as this control approach requires a certain amount of time to converge to an acceptable solution, a common approach consists in pre-training the DRL agent offline with a simulated model of the controlled system, losing the intrinsic model-free nature of the algorithm. Conversely, a DRL controller directly deployed in the controlled environment learning the control policy online may achieve a sub-optimal performance in the first period of deployment, as shown in this study, but can converge to a near-optimal strategy in an acceptable amount of time (in the order of a few weeks as shown in the results). This approach, differently from 827 the DRL with offline training, is model-free in the entire deployment process. These considerations open several research questions on the development of DRL algorithms. If DRL control strategies are implemented with offline 830 training, they require a model of the system, removing this theoretical ad- vantage in comparison to an MPC approach. DRL has the advantage of not relying on a numerical optimization process which generally requires linearized models and and a convex problem 834 to be formalized. This also leads to lower computational times compared to an MPC approach. On the other hand, MPC demonstrated to be a more robust and stable control approach. The flexibility shown by DRL agents is associated with the possibility of temporary poor control performance. This is particularly evident when employing a DRL agent trained online, but this represents nevertheless a promising truly model-free approach. The DRL agent trained online presented in this study proved to be able to improve its control performance over time, approaching the behaviour of a near-optimal MPC strategy or the similar one of a DRL pre-trained offline. However, the possibility to really deploy such a controller in a plug-and-play fashion is still to be assessed, as the hyperparameters and reward function, which play a key role in determining the performance of this category of con- troller, can require different setting depending on the system on which they are implemented. Future work is therefore expected to cover the following aspects:

- Continuing the development of online-trained DRL approaches, by iden- tifying optimal hyperparamenters and reward function configurations that guarantee a fast convergence to a stable control policy, and by

including domain expertise to guide the initial exploration phase.

- Exploring the implications of implementing such advanced control strate gies on more complex case studies, benchmarking and critically dis- cussing the performance of different control approaches.

• Analyzing the capability of DRL and MPC control approaches to adapt to changing environments without the need of external support from a technician. • Implementing a similar benchmarking approach of these control approaches on a experimental setup, providing a more realistic evaluation required for an industrial implementation.

REFERENCES:

1. G. Martinopoulos, K. T. Papakostas, A. M. Papadopoulos, A comparative review of heating systems in eu countries, based on efficiency and fuel cost, Renewable and Sustainable Energy Reviews 90 (2018) pp. doi:https://doi.org/10.1016/j.rser.2018.03.060.
2. A. Kathirgamanathan, M. De Rosa, E. Mangina, D. P. Finn, Data-driven predictive control for unlocking building energy flexibility: A review, Renewable and Sustainable Energy Reviews 135 (2021) 110120. doi:https://doi.org/10.1016/j.rser.2020.110120.
3. R. May, The reinforcement learning method : A feasible and sustainable control strategy for efficient occupant-centred building operation in smart cities, Dalarna University (2019). URL: https://www.diva-portal.org/smash/get/diva2:1358130/FULLTEXT02, (accessed October 14, 2021).
4. T. I. Salsbury,A SURVEY OF CONTROL TECHNOLOGIES IN THE BUILDING AUTOMATION INDUSTRY, IFAC Proceedings Volumes 38 (2005) pp. 90–100. doi:https://doi.org/10.3182/20050703-6-CZ- 1902.01397.
5. M. Molina-Solana, M. Ros, M. D. Ruiz, J. G´omez-Romero, M. Martin Bautista, ata science for building energy management: A review, Renewable and Sustainable Energy Reviews

70 (2017) pp. 598–609. doi:https://doi.org/10.1016/j.rser.2016.11.132.

6. A. Capozzoli, M. S. Piscitelli, S. Brandi, D. Grassi, G. Chicco, Automated load pattern learning and anomaly detection for enhancing energy management in smart buildings, Energy 157 (2018) pp. 336–352. doi:https://doi.org/10.1016/j. energy.2018.05.127.