Model Predictive Controller (MPC)

Robust Model Predictive Controller (RMPC)

Neural Network (NN)

Deep Learning (DL)

Mode Locked Fiber Laser (MLFL)

Varational Autoencoder (VAE)

Model Predictive-Recurrent Neural Network (MP-RNN)

Nonlinear Model Predictive Control (NMPC)

Query By Committee (QBC)

Kullback-Leibler (KL)

Expected Error Reduction (EER)

Literature Review

\*

In the article titled “Learning an Approximate Model Predictive Controller with Guarantees” (Hertneck et al. 2018), a framework based on a Model Predictive Control with Neural Network (NN) was proposed. The framework proposed was based on supervised learning mechanism which was used to approximate MPC “with reduced computational complexity and guarantees on stability and constraint satisfaction” (Hertneck et al. 2018). The main reason to incorporate Model Predictive Controller with a neural network controller was that MPC contains “computational effort” while tackling “optimization problems **online** under **real-time requirements**” (Hertneck et al. 2018).

The designed proposed in the paper starts off with a Robust Model Predictive Controller (RMPC) which emphasis on containing the “robustness to bounded input disturbances” (Hertneck et al. 2018). The results obtained is sampled and approximated with suitable approximation techniques also known as regression techniques: neural networks etc. The logic was further built and within the admissible region of the RMPC, the stability was guaranteed for the closed-loop system. It was also suggested that the applicability of such frameworks is not just limited to linear systems. It can target range of nonlinear control problems with any approximation function (regression) technique.

As the system was complex, following assumptions were made in designing of the Robust Model Predictive Controller (RMPC):

“Assumption 1: (Local incremental sterilizability

Assumption 2: (Local Lipschitz continuity)

Assumption 3: (Bound on the input disturbance)

Assumption 4: (Terminal set)” (Hertneck et al. 2018).

The use of neural networks as approximators has been found really effective. These neural networks can approximate regular function to “arbitrary accuracy provided that the network has sufficiently many hidden units” (Hertneck et al. 2018). Furthermore, these neural networks can be used with statistical learning to keep the validation bound within its limits.

\*

The understanding from the article named “Deep Learning and Model Predictive Control for Self-Tuning Mode-Locked Lasers” is based on the implementation of a Model Predictive Control with the infamous machine learning methods to have a more reliable self-tuning system. The idea is propelled on the mass accumulation of data and the rise in data driven modelling of physical systems. The use case for instance can be found in “optical systems, including optical communication, laser cutting, metamaterial antenna arrays, microscopy, and characterization of x-ray pulses” (Baumeister, Brunton and Nathan Kutz 2018).

The implementation of the Model Predictive Control (MPC) and the Deep Learning (DL) Algorithms were executed on Mode Locked Fiber Laser (MLFL). The MLFL was coupled with a Varational Autoencoder (VAE), a latent variable mapping and a model predictive controller. The objective function (O) in this case was Energy Function (E) divided by the fourth-moment (kurtosis). That is,

M of the Fourier spectrum of the waveform

O = E/M

While training the model, the two main challenges that occurred were “(i) the exploding and the vanishing gradient effects, which are especially critical for problems with long-range dependencies, and (ii) the effect of nonlinearities when iterating from one time step to another” (Baumeister, Brunton and Nathan Kutz 2018). Thought difficult, the authors worked on various strategies to combat the challenges. Firstly, the data set was divided into equidistant parts and named it Ksubset. Furthermore, the Deep Learning Control Algorithm was based on two loops: inner and outer. The inner loop, on one hand had the Varational Autoencoder (VAE) and the K -***a*** mapping whereas, the outer loop had the Model Predictive-Recurrent Neural Network (MP-RNN). The resulted architecture was able to approximate the unknown elements in fiber birefringence, built a model that is dynamical in nature and can “appropriate control law for maintaining robust, high-energy pulses despite a stochastically drifting birefringence” (Baumeister, Brunton and Nathan Kutz 2018).

In the field of controller design, machine learning and artificial intelligence have a yet to apply crucial role and will be the hot topics in the upcoming industrial projects. In the case of Mode-Locked Lasers, the paper serves as a stepping stone for the robust and stable methods for achieving self-tuning performance in the Mode Locked Fiber Laser (MLFL).

\*

Article: Extension of the do-mpc Development Framework to Real-time Simulation Studies

The article titled “Extension of the do-mpc Development Framework to Real-time Simulation Studies” roots the idea of shifting from the theoretical to an application side vis-à-vis to the implementation of Nonlinear Model Predictive Control (NMPC). Though frequently used in the research sector, industrial applications are yet to me commercialized to validate the NMPC results in an actual real-time simulation environment. Thus, this article specifically proposes an extension in software environment of do-mpc framework to facilitate from a synchronous to asynchronous software structure to support the real-time scenarios of the Nonlinear Model Predictive Control (NMPC) controllers.

In the research side, NMPC have been considered a lot in applications such as distillation column, bed process and reactive distillation column. It has shows great results dealing with the non-linearity in the models with reliable outcomes. As the advancement in the technologies and the emergence of Industry 4.0, the implementation of NMPC is possible with a variety of tools and platforms. For the successful implementation of the NPMC, the article states that:” the controller and the NMPC solution should be validated in a real-time simulation environment before they are applied to the real plan” (Tatulea-Codrean et al. 2019). Hence, this research article describes development phases: Synchronous Simulations, Asynchronous Simulations and Plant Implementations, and the need for the model to have parallel computation to simulate real-time environment.

In order to work with the real-time simulations, do-mpc is extended on software basis. The software extension integrates “both simulation modes in an easily configurable software structure” (Tatulea-Codrean et al. 2019). The communication principles in the Synchronous, Asynchronous and Plant Implementation based on the collection of data via Database. The modules in Asynchronous and in the Plant Implementations writes the calculated data and saved it to the server where the software stores it regularly. The data stored is readily available in case it is needed in the future. In the process of real-time simulations, feedback delays occur. In the case of NPMC proposed, the Real-time iterations is used to approach the feedback delay, constraint violations and performance losses.

(Tatulea-Codrean et al. 2019)

\*

Active Learning also known as “Query Learning” and “Optimal Experimental Design” is a field driven from Machine Learning, specifically Artificial Intelligence. The Idea behind the technique is to limit the use of resources in labelling the data points. In Active learning, an intelligent algorithm can interact a *teacher* or *oracle* to label data points in relevance to the desired output as labelling instances, for a large sum of data, which can be tedious, time consuming, difficult and/or expensive, depending on the labelling situations. Active Learning is useful in areas of Speech Recognition, Information Extraction, and Classification and Filtering of data.

To label unlabeled data points, Active Learning requires the expertise of an Oracle, also known as a *teacher* which could be a human and/or trained algorithm. The system of Active Learning generates queries in the form of unlabeled instances which are eventually be labeled via an Oracle who possesses sufficient knowledge and expertise to validify the decision. The new labeled data is transferred back to the machine learning model and the cycle repeats till the required accuracy is met. This is called The Pool-Based Active learning Cycle.

“Active learning algorithms are *generally* evaluated by constructing learning curves, which plot the evaluation measure of interest” (Settles 2012). Although, the evaluation of the Active Learning varies to the case and scenario, constructing learning curves may help in situation where label classification is low.

The report titled: “Active Learning Literature Survey” talks about three main scenarios vis-à-vis problems in which Active Learning System can ask queries to the oracle. The three main query asking scenarios are:

1. Membership Query Synthesis
2. Stream-Based Selective Sampling and,
3. Pool Based Active Learning

The usage of the scenarios depends on the data type, data query decision type, time limitation.

Query By Committee (QBC) is a selection algorithm in the active learning domain. In this selection algorithm, there is a committee of models also known as classifiers () which are trained on Labels’ set (). These models differ among each other based on the competing of hypothesis criteria (agreement and disagreement among models). When a query looks for it label, the committee votes on its probability and rank it among the queue based on from the best to the worst query.

In order to implement Query by Committee Algorithm, following steps needed to be taken:

* Construct the committee of models which should represent some area of the Version Space.
  + Query By Bagging and/or
  + Query By Boosting
* Devise a methodology to measure the disagreement among committee
  + Vote Entropy
    - where ranges over all possible labelings, and is the number of “votes” that a label receives from among the committee members’ predictions.
  + Kullback-Leibler (KL) divergence
  + Jensen Shannon Divergence

The QBC strategy is prone to querying outliers.

The other Query Strategy Frameworks included in this literature are:

* Uncertainty Sampling,
* Expected Model Change,
* Variance Reduction and Fisher Information Ratio,
* Estimated Error Reduction and,
* Density-Weighted Methods

Variance Reduction and Fisher Information Ratio is an advance function that helps in statistical analysis of active learning. The Expected Error Reduction (EER) directly minimizes the loss function which poses two main challenges:

* The model needs to be retrained with the trained tuple in a loop,
* Minimize the future risk.

In Variance Reduction, the indirect approach is used to minimize the loss function by minimizing the output variance in a closed loop from a loss function. The mathematical calculation in a closed form is in a finite number of steps which leads tangible and a feasible result.

The Expected Future generalization error is as follows:

Where;

is an expectation over some labeled set of a given size, is an expectation over the conditional density, and is an expectation over both cases.

In neural networks, the approximated output variance is calculated using Fisher Information Matrix stated in the following approximated equation:

Where;

is the mean squared error of the current model on the training set.

Minimizing the variance helps to minimize the error in the model. It then “query the instance resulting in the greatest future variance reduction” (Settles 2012):

The indirect approach to the variance is feasible in the closed-form as compared to the Expected Future Generalization Error.

\*

In the early machine learning processes, the algorithm was mainly driven in the offline format adhering to the static input of data. In the traditional machine learning process, following methods were taken to complete and update the machine learning algorithm:

* Receiving Static Input data file,
* Perform Exploratory Data Analysis,
* Perform Feature Engineering,
* Dividing the data into Training, Testing (for fine tuning the model parameters) and Validation (model selection) sets,
* Deployment of the trained model set to the production.

(Jeeva 2019).

Online Machine Learning also known as Incremental Machine Learning which generates sequences of models () on a stream of training data (). It is also important to understand that the model has its dependency on the parameters obtained from the previous model and the recently obtained parameters updated from the data received in the stream.

The algorithms suggested in this article regarding on how to update the trained model parameters and to have the improved and enhanced model performance metrics are Scikit-learn, Warm start a Neural Nets and Bayesian methods are well suited.

\*

\*

**Reference Style: Chicago**

Hertneck, Michael, Johannes Kohler, Sebastian Trimpe, and Frank Allgower. 2018. "Learning An Approximate Model Predictive Controller With Guarantees". *IEEE Control Systems Letters* 2 (3): 543-548. doi:10.1109/lcsys.2018.2843682.

(Hertneck et al. 2018)

Baumeister, Thomas, Steven L. Brunton, and J. Nathan Kutz. 2018. "Deep Learning And Model Predictive Control For Self-Tuning Mode-Locked Lasers". *Journal Of The Optical Society Of America B* 35 (3): 617. doi:10.1364/josab.35.000617.

(Baumeister, Brunton and Nathan Kutz 2018)

Tatulea-Codrean, Alexandru, Clemens Lindscheid, Rafael Farrera-Saldana, and Sebastian Engell. 2019. "Extension Of The Do-Mpc Development Framework To Real-Time Simulation Studies". *IFAC-Papersonline* 52 (1): 388-393. doi:10.1016/j.ifacol.2019.06.093.

(Tatulea-Codrean et al. 2019)

Settles, Burr. 2012. "Active Learning". *Synthesis Lectures On Artificial Intelligence And Machine Learning* 6 (1): 1-114. doi:10.2200/s00429ed1v01y201207aim018.

(Settles 2012)

Jeeva, Manikandan. 2019. "Data Streams And Online Machine Learning In Python". *Medium*. <https://medium.com/analytics-vidhya/data-streams-and-online-machine-learning-in-python-a382e9e8d06a>.

(Jeeva 2019)