Directed Study Course Report (ENSC 891)

Title: Learning-Based Model Predictive Control (LBMPC)

Instructors: Kamal Gupta, Mehran Mehrandezh

Student: Abolfazl Eskandarpour[1]

September, 2020

[1]Department of Engineering Science, Simon Fraser University, e-mail: eskandar@sfu.ca

Contents

1 Introduction

In optimal control design for complex nonlinear systems, one of the main concern is the existence of trade-off between robustness and performance of the closed-loop system. The aforementioned trade-off can be difficult to deal with if a poor system model would be identified while state and input constraints are imposed on the system. So, one strategy to cope with this problem is to devise a framework to decouple the performance and robustness of the system during the optimization process as much as can be achievable. In this regard, trends towards integrating machine learning methods and optimal controller strategies are increasing.

Machine Learning Control (MLC) is a sub-field of machine learning, intelligent control, and control theory which solves optimal control problems using machine learning methods. The main applications are complex nonlinear systems for which linear control theory methods are not applicable. In fact, machine learning techniques are widely used in learning and data-driven control techniques. So, their capabilities in system identification and parameter estimation make them as powerful tools for optimal control problems. On the other hand, amongst optimal control strategies, Model Predictive Control (MPC) can deal with model uncertainties in constrained complex systems properly by handling the practical constraints imposed on the system. Another advantage of MPC over the other optimal controllers is the fact that it can optimize the problem by keeping both current and future timeslots into account. In addition, there are a wide range of approaches in the literature by which MPC can guarantee the feasibility, stability and robustness of the closed-loop system. The combination or integration of the MPC and machine learning methods can be identified as Learning-Based MPC (LBMPC) in the literature.

Learning Based Model Predictive Control (LBMPC) scheme not only can provide deterministic guarantees on robustness, but also can identify a more accurate model for the system to increase the performance of the closed-loop system [1]. The main merit of LBMPC is that robustness and performance can be decoupled under reasonable conditions in an optimization framework by maintaining two models of the system. First model is an approximate one with by considering the uncertainties on that, and the second model can be updated by statistical methods [1]. LBMPC can improve the performance

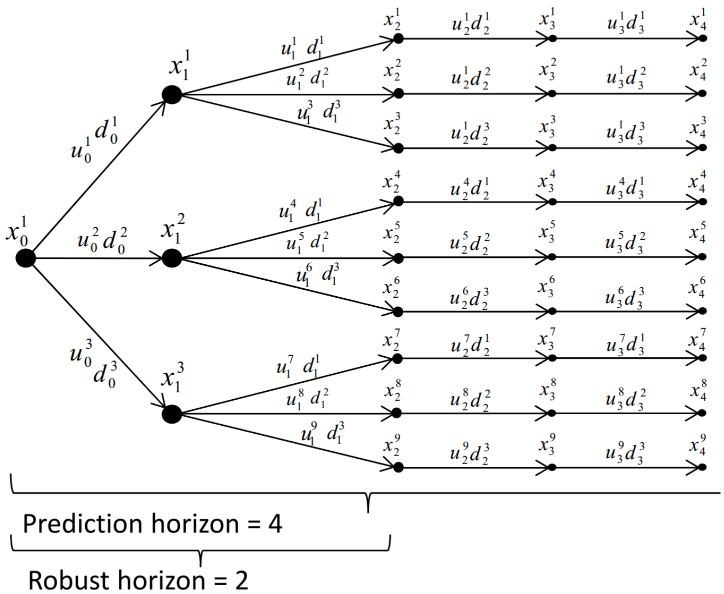

Figure 1: Tree representation of the evolution of the uncertainty for NMPC [7]

of system by minimizing the cost function and finding corresponding inputs subject to the learned dynamics, and it ensures safety and robustness by checking whether these same inputs keep the approximate model stable when it is subject to uncertainty [1].

LBMPC approaches can be considered in three main categories which will be addressed more in the following sections. In the first approach which is addressed in most researches, the prediction learned model can be automatically improved using the recorded data. As the second approach, as stated already, there are some techniques to improve the closed-loop performance of the system by working on the MPC parameters related to the cost and constraints of the objective function. It is worth mentioning the there is a increasing interest towards the second approach. Last but not least, the MPC is used to augment learning-based controllers with constraint satisfaction properties [2].

The rest of the report is organized as follows. In ”Literature Review” section the recent papers on the LBMPC approach in the literature is discussed. In "Background and Methods" section, the theories behind the Learning-Based controllers and MPC controller are addressed separately to give a clear insight to the reader that which parts of the MPC would be modified to contain the concept of the Learning approaches. In "Examples and Simulation" section, some simulation related to LBMPC methods are provided. Finally, to distinguish the proposed methods with each other, the results are discussed and concluded in the "Discussion and Conclusion" sections, respectively.

2 Literature Review

The MPC controller is a model-based controller and the performance of the closed-loop system rigorously depends on the descriptive model of the system subject to the practical constraints. In this regard, model uncertainty is a prevalent issue in the context of MPC which causes feasibility problems which considerably affects the performance of closed-loop system, and also it is probable that the system would be unstable under some circumstances like external disturbances. Although MPC preserves some degree of robustness to sufficiently small uncertainties due to its prediction nature of implementation, a marginal robust performance may not be adequate in many practical situations. In this regard, a wide range of works have been done on designing robust, adaptive and stochastic MPC controller to deal with such problems [references related to robust MPC]. Uncertainty in MPC is handled by optimizing over multiple uncertain forecasts. In this case, performance index and operating constraints take the form of functions defined over a probability space, and the resulting technique is called stochastic MPC [3].

Yet the modified MPC cannot actively learn about the system uncertainties. In particular, the MPC cannot proactively keeping up with system changes that may occur due to the time varying nature of system dynamics, which can in turn increase uncertainty even more, leading to control performance degradation [4]. Solving the root of the problem, having a better model identification for descriptive model can alleviate the trade-off between performance and robustness of the closed-loop system. This can be done by employing some techniques to analyze collected data of the system before and during operation [5].

Machine learning has drawn increasing level of attention in model identification recent years. Its combination with MPC (LBMPC) handles system constraints, optimizes performance with respect to a cost function and uses statistical identification tools to learn model uncertainties [6]. LBMPC uses a linear model with bounds on its uncertainty to construct invariant sets that provide deterministic guarantees on robustness and safety. An advantage of LBMPC is that many types of statistical identification tools can be used with it to cope with model uncertainty in the system, such as Recurrent Neural Network (RNN), Deep Newural Network (DNN) and Reinforcement Learning (RL) [2].

The merits of using LBMPC go beyond just dealing with model uncertainty and a data-driven model improvement. In fact, it can represent the relationship between the specific parameterization of an MPC controller such as the cost function, control and prediction horizon, constraints and also weighting matrices, which cannot be analytically realized [7]. For example, some researchers combine MPC and RL in the context of LBMPC to investigate constraint satisfaction and feasibility, while closed-loop performance is optimized [8]. The learning part of LBMPC can be used either online or offline. In the offline strategy, learning-based methods are employed to approximate explicit MPC control laws [9] [10] [11] [12]. In the online method, the learning and optimization phase of the LBMPC work simultaneously and interactively at each sampling time [8]. One of the main concern is the trade-off between learning and optimization processes of LBMPC which can be defined in the area of dual control [4]. In fact, maintaining optimal balance between identification and control parts of LBMPC is difficult since they are naturally in conflict. In this report, the dual control affect is not addressed in detail and just the LBMPC identification and control parts are investigated.

Considering the time that learning techniques are applied to the MPC controller, LBMPC can be divided into two category, offline and online learning. For instance in the offline learning methods, to avoid the controller complexity, the data required for mapping between the states of the system and the robust optimal control inputs are generated [13]. In contrary, in online learning methods, the controller parameters is adjusted during closed-loop operation or using the data collected from previous and current controller

Table 1: Organization of references based on the proposed topics

Topic	Subtopic	Reference
Learning the System Dinamics	Robust models	15, 16
	parametric	20, 21
	non-parametric	22, 23
	Stochastic models	24, 25
	parametric	26
	non-parametric	27
Learning the Controller Design	Performance-driven learning	-
	Bayesian optimization	28
	Terminal components	29
	Inverse optimal control	30
MPC for Safe Learning	-	7, 26, 31

iteration. A lots of research have been done on identifying the model to have a better description of that, several research efforts are addressing the formulation of the MPC problem to satisfy constraints and to improve the closed-loop performance during learning-based control [14]. Based on the structure that the learning methods take place and alter the MPC controller, the research efforts towards LBMPC can be categorized into three main approaches: learning the system dynamics, learning the controller design and MPC for safe learning [2].

3 Background and Methods

3.1 Learning the system dynamics

In conventional MPC the nominal system model is derived offline based on physical principles and applying some system identification techniques. In conventional stochastic MPC the uncertainties are considered as a distributional information, while in robust case the uncertainties and disturbances are supposed to lie in a predetermined set like $\theta \in T$ and $w(k) \in W$, respectively. Both conventional approaches set the model and uncertainty description before the design of the controller. In contrary, the learning part of LBMPC uses state measurements x (past and current) and update uncertainties and disturbance sets during operation at each samoling time k to find the controller policy u. The system dynamic in LBMPC can be written explicitly as a summation of nominal system model f_n and learned term f_l as follows [15]:

$$f(x, u, k, \theta, w) = f_n(x, u, k) + f_l(x, u, k, \theta, w) \tag{1}$$

Since there is a rich theory behind the robust and stochastic MPC, those approaches have been extended into the proposed category of learning. So, those learning methods can be organized into robust and stochastic LBMPC. It is worth mentioning that, in robust MPC or LBMPC controller design, the uncertainty parameter θ is stochastic. So, one can say that the system is robust in probability. On the other hand, the second approach can be called fully stochastic in which the conservatism in designing controller (like robust MPC or LBMPC) has been reduced [2]. The robust and stochastic LBMPC are addressed in sections 3.1.1 and 3.1.2, respectively.

3.1.1 Learning the system dynamic using robust concept

Considering Equation 1, the uncertainty is assumed to lie in a compact set $f_l(x, u, k, \theta, w) \in W$, and the controller is designed to be robust against the additive uncertainty. For the nominal system model f_n some learning-based methods are used for a performance model that is generated or improved from data. So, with this approach, while the feasibility can be guaranteed by robust MPC, the closed-loop performance of the system can be improved using the data during the operation [16]. In [16], the authors employ the nominal and a learned performance model, respectively as follows:

$$z(k+1) = Az(k) + Bv(k) + w(k), \tag{2}$$

$$x(k+1) = Ax(k) + Bu(k) + O(x, u, k) \tag{3}$$

where $O(x, u, k)$ is an oracle representing an arbitrary learning-based approximation of $f_l(x, u, k, \theta, w) \in W$. The practical implementation of this strategy can be found in [17] [18] [19]. In the proposed method in [20], the model $fl(x, u, k, \theta, w) \in W$ is fixed beforehand which is called parametric approach. Many of techniques are based on the set membership identification.

On the other hand, in [21], they approximate $f_l(x, u, k, \theta, w) \in W$ based on the observed data which this technique is non-parametric one.

3.1.1.1 Robust parametric approach Considering the measured state X and input trajectory U, the parametric set-membership T_k is a set of θ by which the observed state X and input trajectory U are consistent, so we have [2]:

$$T_k = \{\theta | \forall j = 0, ..., k \; \exists \; w \in W, such\ that\ x(j+1) = f(x(j), u(j), j, , w)\} \quad (4)$$

The next update would be:

$$T_{k+1} = \{\theta \in T_k | \exists \; w \in W, such\ that\ x(k+1) = f(x(k), u(k), k, , w)\} \quad (5)$$

So, we can see that:

$$T_{k+1} \subseteq T_k \quad (6)$$

which means the set of consistent parameters is non-increasing over time.

3.1.1.2 Robust non-parametric approach As already stated, in the robust non-parametric approach, there isn't any explicit description of f_l based on x, u, k, θ and w. based on the observed data which this technique is non-parametric one. In this approach, researchers use Lipschitz interpolation [22] and kinky inference [23] methods to form a strict bounds for f_l. Again, considering the measured state X and input trajectory U, the feasible system F_k can be defined as:

$$F_k = \{f || \bigtriangledown f| \leq \epsilon, \forall 0 \leq j \leq k \; \exists \; w \in W, such\ that\ x(j+1) = f(x(j), u(j)+w)\} \quad (7)$$

The next update would be:

$$F_{k+1} = \{f \in F_k || \bigtriangledown f| \leq \epsilon, \forall 0 \leq j \leq k \; \exists \; w \in W, x(k+1) = f(x(k), u(k)+w)\} \quad (8)$$

In general, a nominal function f might be chosen between F_k and F_{k+1} to increase the distance between them.

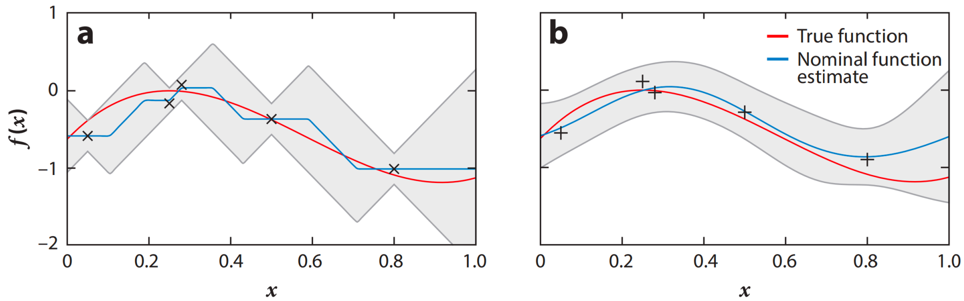

Figure 2: Robust and stochastic nonparametric estimation techniques. Noisy measurements of the true function (red) are shown as crosses; the nominal function estimate is shown in blue. a)Robust set-membership technique b) Gaussian process regression [2]

3.1.2 Learning the system dynamic using stochastic concept

In stochastic LBMPC, in contrary to robust LBMPC, it is no longer need to have a strict and conservative bound on all uncertainties. In fact, it uses distributional information on the system to design the controller [24]. The difference between robust and stochastic approach is that the uncertainty parameter changes from deterministic notion stochastic one, i.e. we have the parameter distribution $\rho(\theta|X,U)$ which explains the knowledge about the system dynamic given data X and U. Here, there is a feasible parameter set T_k like equation (4) that in which $Pr(\theta \in T_k|X,U) \geq \rho$ [Learning-based robust model predictive control with state-dependent uncertainty].

As stated in [25], the robust method can be applied into stochastic one by designing a controller which is robust in probability. Unfortunately, there are two issues which might happen. First, the assumption in the stochastic method that the disturbance $w(k)$ is Gaussian, sometimes it can be unbounded which is against with the robust method assumptions. Authors in [26] cope with this problem by separating the robust probability and stochastic parts which deal with parametric uncertainty and unbounded noise, respectively. Second, based on the equation (6) $T(k+1)$ should lie in $T(k)$ as iteration goes by which isn't compatible with the unbounded feature of a stochastic parameter. To deal with aforementioned problems, similar to robust method, the stochastic method can be split into parametric and non-parametric ones.

3.1.2.1 Stochastic parametric approach

To estimate the uncertainty parameter, the posterior distribution of that given X and U can be written based on Bayes' rule as follows [2]:

$$\rho(\theta|X,U) = \frac{\rho(X|U,\theta)\rho(\theta)}{\int \rho(X|U,\bar{\theta})\rho(\bar{\theta})d\bar{\theta}} \tag{9}$$

The recursive parameter update can be written as:

$$\rho(\theta|x(k+1),u(k),X,U) = \frac{\rho(x(k+1)|x(k),u(k),\theta)\rho(\theta|X,U)}{\int \rho(x(k+1)|x(k),u(k),\bar{\theta})\rho(\bar{\theta}|X,U)d\bar{\theta}} \tag{10}$$

So, the posterior distribution can be updated without using the entire state and input history.

3.1.2.2 Stochastic non-parametric approach

For non-parametric approaches, Gaussian process (GP) regression is commonly used as learning approach by directly providing an assessment of the residual model uncertainty after learning [27]. GP regression for model learning usually assumes dynamics with independent and identically distributed, zero mean additive Gaussian noise $w(k) \sim N(0, \sigma_w^2)$ of the form $x(k+1) = f(x(k), u(k)) + w(k)$ and having values of the true function dynamic at different inputs x, u are jointly Gaussian distributed according to a kernel function k, expressing the covariance between function values [2].

3.2 Learning the controller design

Although an accurate representation of system model has more effect on the performance of the closed-loop system, the cost function, the constraints on states and inputs and weighting matrices can considerably affect the performance of the system. A parameterized MPC can be written as follows:

$$
\begin{aligned}
&U^\star = \arg \min_{U} \quad \sum_{i=0}^{N} l(x_i, u_i, \theta_l) \\
&\quad \text{subject to :} \quad x_{i+1} = f(x_i, u_i, \theta_f), \\
&\quad U = [u_0, ..., u_N] \in \mathcal{U}(\theta_u), \\
&\quad X = [x_0, ..., x_N] \in \mathcal{U}(\theta_u), \\
&\quad x_0 = x(k).
\end{aligned} \tag{11}
$$

This LBMPC approach can be split into two categories. First, Performance-driven controller learning is employed to improve the closed-loop performance by successively adjusting the parameters in Equation 11. Second, Learning from demonstrations via inverse optimal control can observe a desired behavior of a closed-loop control system to design an automatic controller. With this approach the parameters of Equation 11 can be obtained using recorded data.

3.2.1 Performance-Driven Controller Learning

MPC controller in fact is a rough approximation of the true stochastic optimal control problem because of its finite prediction horizon, simplified models, and since stochastic disturbances is neglected in its design. There are two ways to deal with this problem. First, the Bayesian optimization is widely employed to optimize $J(\theta)$ the objective function using GP regression as a regularity assumption to finally optimize the true objective function $J_t(\theta)$. Based on this estimate of the function $J(\theta)$, an acquisition function $\alpha(\theta)$ is used to trade off exploration and exploitation, and to determine $\theta = arg\ min_\theta\ \alpha(\theta)$ as the next parameter to be evaluated [28]. Second, finding the large terminal set using learning methods is addressed to improve the performance of the closed-loop system. Moreover, this approach help to find a large terminal invariant set to ensure recursive feasibility for a large set of initial states to guarantee asymptotic stability of the closed-loop system [29].

3.2.2 Learning from Demonstration with Inverse Optimal Control

Inverse optimal control addresses the problem of defining an objective function l or constraints X, U systematically by inferring them from demonstrations. To do so, a prediction model f of the system dynamics is usually assumed to be given. By learning the components of an optimal controller rather than imitating an observed policy, inverse optimal control offers favorable generalization properties, such that a resulting MPC control law can be derived for the entire state space. Following steps are used to apply this method [30]:

1. Define the optimal control problem with a parametric cost function $l(x, u, \theta_l)$ costs with unknown weights $l(x, u, \theta_l) = x^T Q(\theta_l)x + u^T R(\theta_l)u$,
2. Derive optimality conditions for the parametric optimal control problem,
3. Solve optimality conditions for the parameters θ_l given the demonstration

3.3 MPC for safe learning

Although most researches have shown the successful capability of learning-based methods in solving complicated control problems, they cannot guarantee the stability due to the physical constraints imposed on the system. The main idea is to decouple the optimization of the objective function from the requirement of constraint satisfaction, which is addressed using MPC techniques. In this regard, a solution is that MPC might be used to turn a safety-critical dynamical system (a safety filter) into an inherently safe system to which any learning-based controller without safety certificates can be applied [26](Figure 3).

The procedure is that at first the stochastic optimal control problem is solved through learning-based control methods, such as stochastic policy search or approximate dynamic programming [7]. Then the solution would be verified in terms of safety by computing a safe backup trajectory from the one-step predicted state $x_{1|k}$ to a safe terminal set X_f.The subtle point is that the proposed procedure to validate safety of the gained input is computationally cheaper than (lower prediction horizon) just solving J_t without learning-based methods [31].

4 Examples and Simulation

In this section some example of Learning methods and MPC from MATLAB MPC and Deep Learning toolbox and Simulink are provided. In the first example, the simulation results show that how the Deep Neural Network (DNN) can imitate a Nonlinear MPC (NMPC) controller for a Flying robot

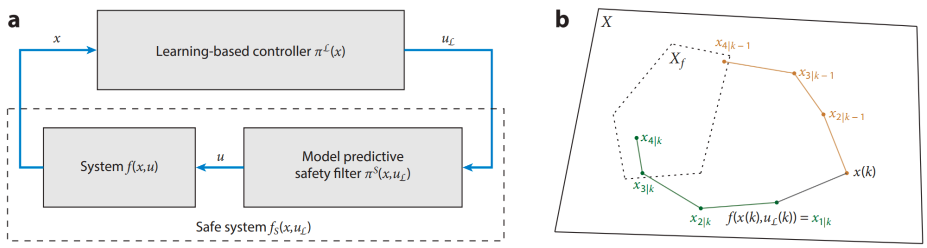

Figure 3: Model predictive safety filter [2]

and a Lane Keeping Assist. In the second example, a combination of Reinforcement Learning and MPC is used for automatic search and parking task. In the third example, a neural network predictive controller is demonstrated to control a Continuous Stirred Tank Reactor.

4.1 Imitating NMPC using DNN

In these examples, the DNN is trained, validated and tested to imitate the behavior of NMPC predictive controller for a Flying robot and a Lane Keeping Assist. Using this technique can be reasonable, since evaluating a DNN can be more computationally efficient than solving a NMPC problem in real-time. After training the network, the DNN network is created by considering the layers in Figure 4. Moreover, the results in Figure 5 and 6 demonstrate the successful imitating DNN of NMPC. Also, Figure 7 shows the needed layers of DNN for Lane Keeping Assist. Figure 8 shows the training time and episodes during which the Lane Keeping Assist is trained. Finally, Figure 9 and 10 demonstrate the comparison of DNN and MPC in state following and steering angle control for the Lane Keeping Assist.

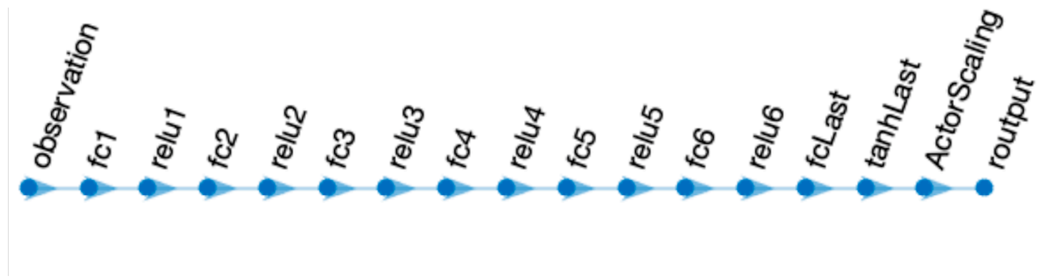

Figure 4: DNN layers for Flying robot

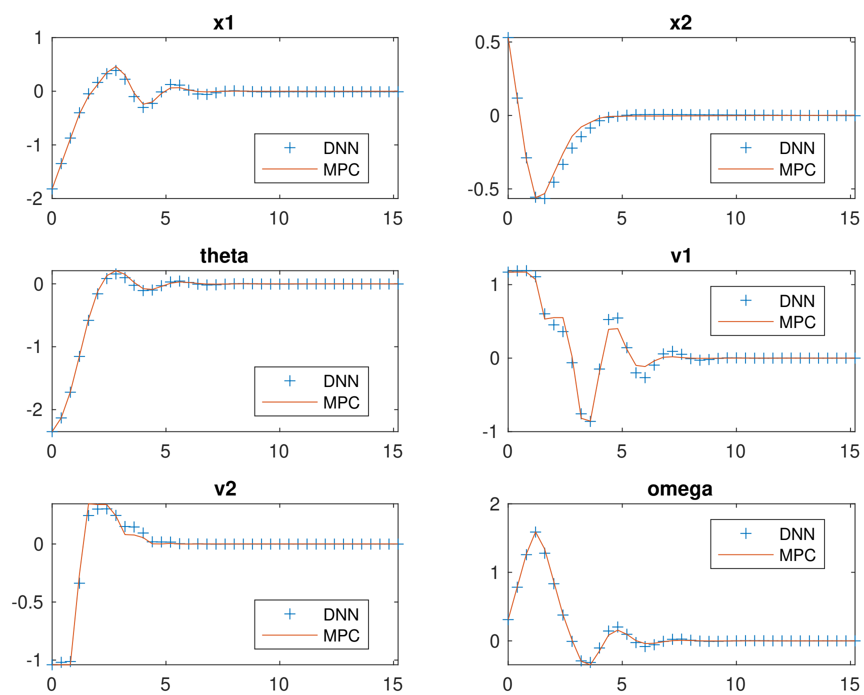

Figure 5: Flying robot state following using DNN and NMPC

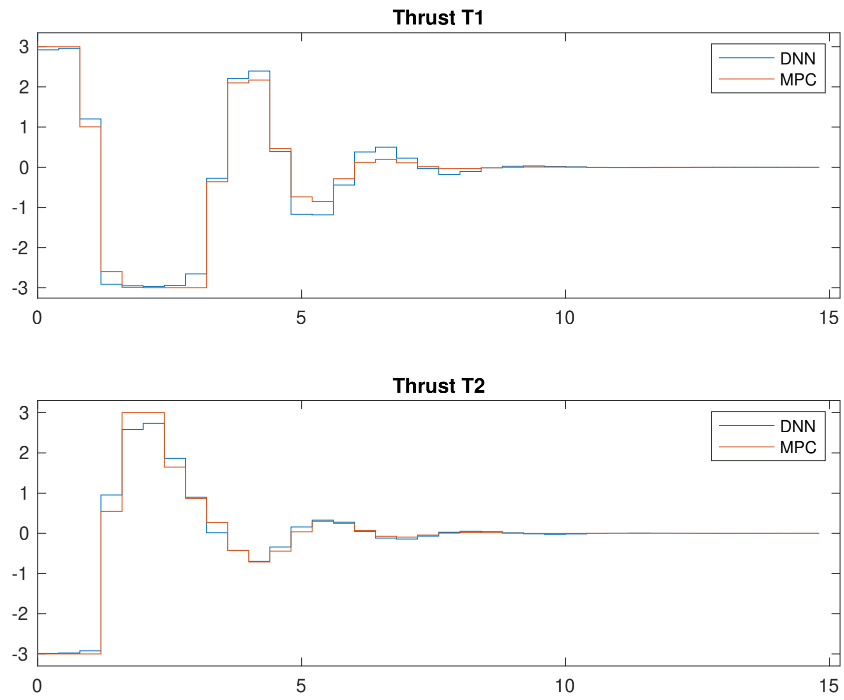

Figure 6: Thrust of Flying robot using DNN and NMPC

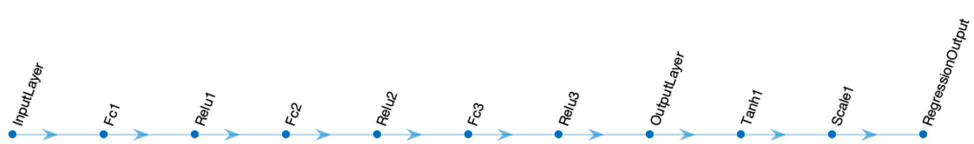

Figure 7: DNN layers for Lane Keeping Assist

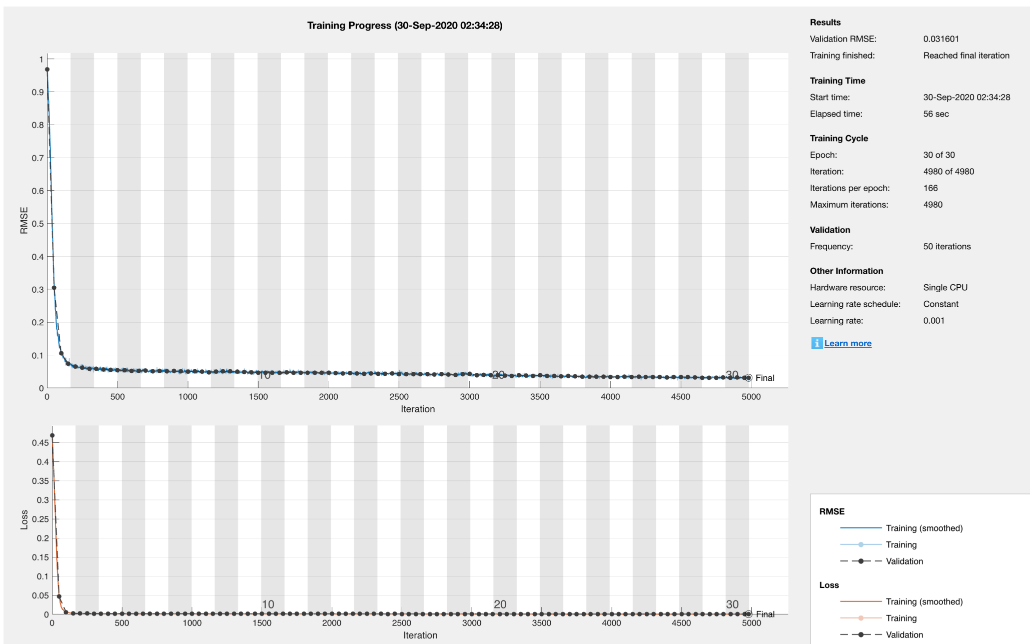

Figure 8: Training Episodes and Time for Lane Keeping Assist

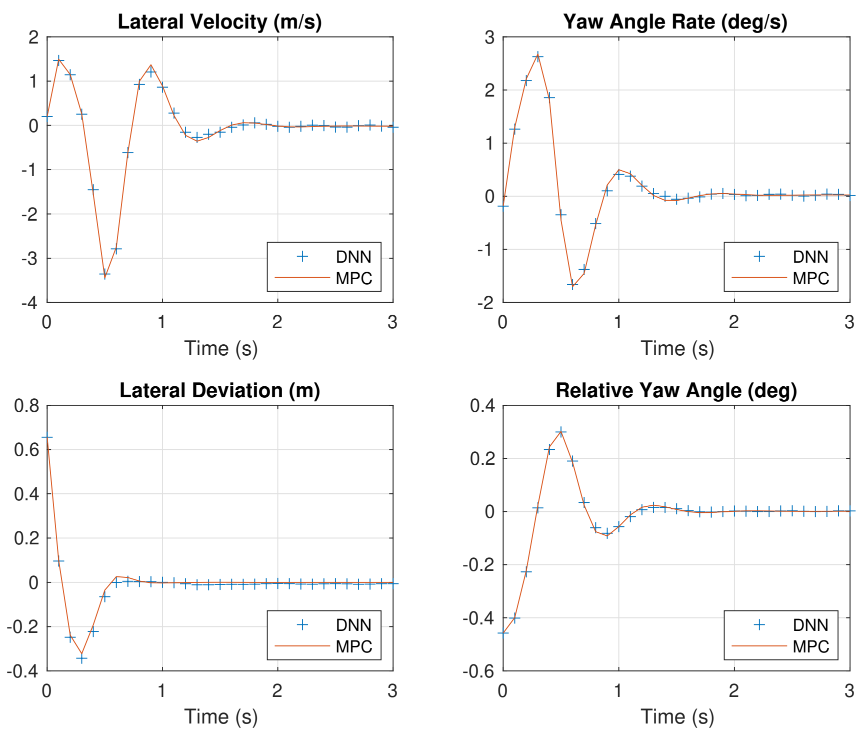

Figure 9: Lane Keeping Assist state following using DNN and MPC

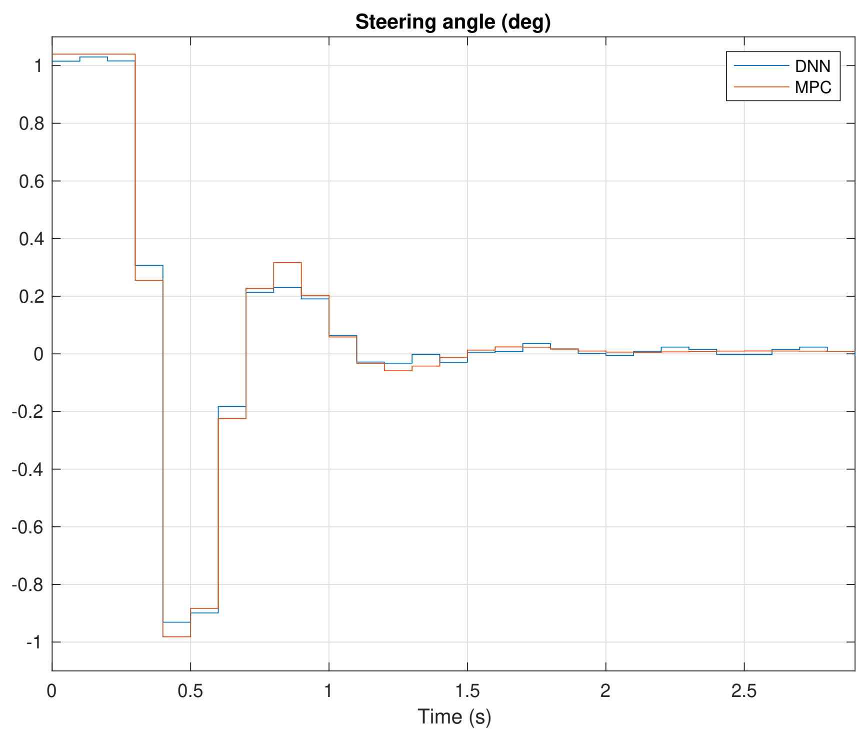

Figure 10: Steering angle of Lane Keeping Assist using DNN and MPC

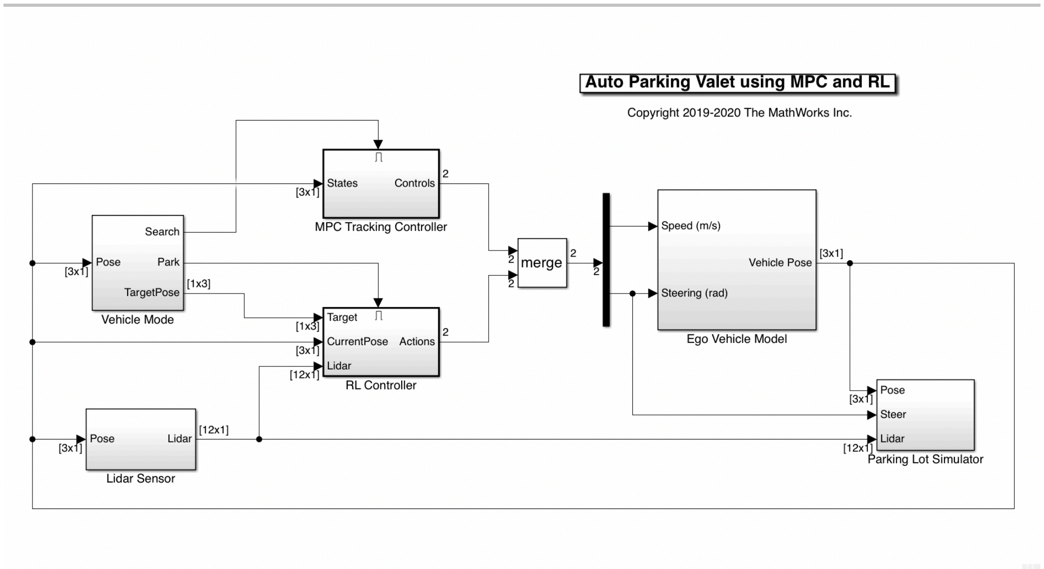

Figure 11: Simulink of RL and MPC scheme for Automatic Car Park

4.2 Reinforcement Learning and MPC

This example demonstrates the design of RL and MPC controller for automatic search and car parking. The hybrid controller employs MPC to follow a reference path in a parking lot and a trained RL agent to perform the parking maneuver. The controller helps the car to avoid obstacles and search for a empty spot for parking. The environment and the location of empty parking spots are known to the controller.

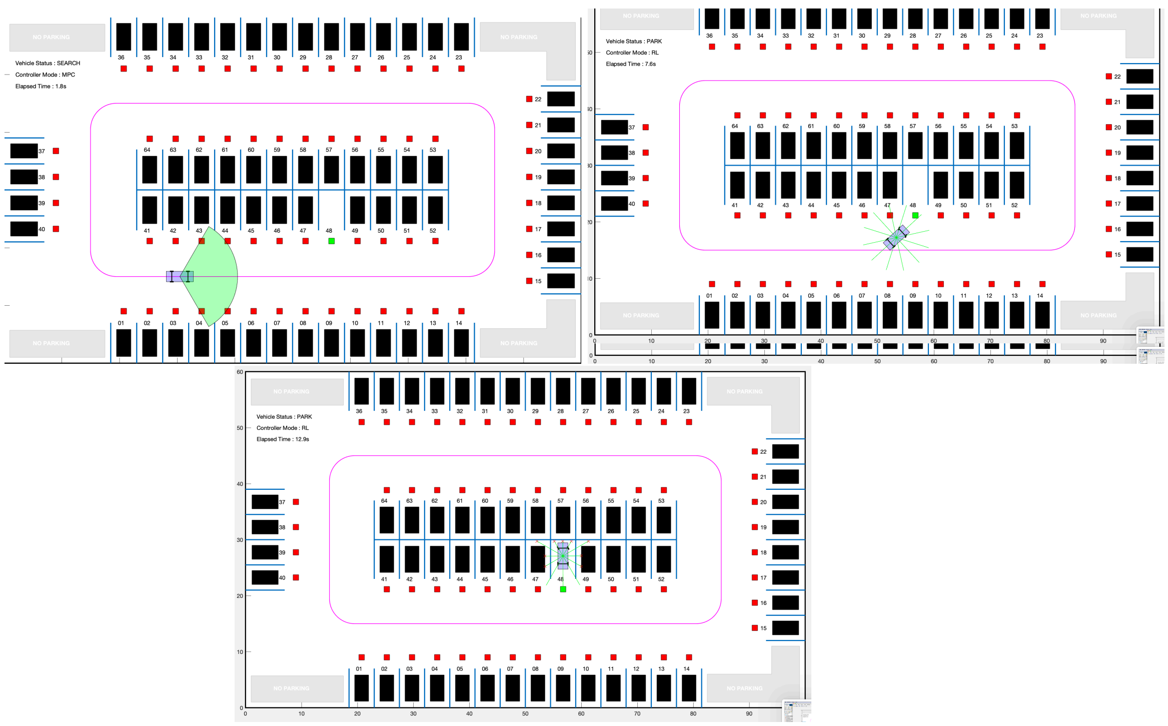

Figure 12: Automatic Car Park Scenario using RL and MPC

4.3 NN Predictive Control

The NN predictive controller employs a neural network model of a nonlinear plant to predict future plant performance. The controller then calculates the control input that will optimize plant performance over a specified future time horizon. The first step in model predictive control is to determine the neural network plant model (system identification). Then, the plant model is used by the controller to predict future performance of the Continuous Stirred Tank Reactor (CSTR). Figure 13 demonstrates the overall scheme of NN predictive controller for CSTR. Figure 14 shows the input-output data for system identification. Figure 15 shows the information regarding the training phase and figure 16 demonstrates the validation and training error. Finally, figure 17 shows the success reference trajectory following for the NN predictive controller.

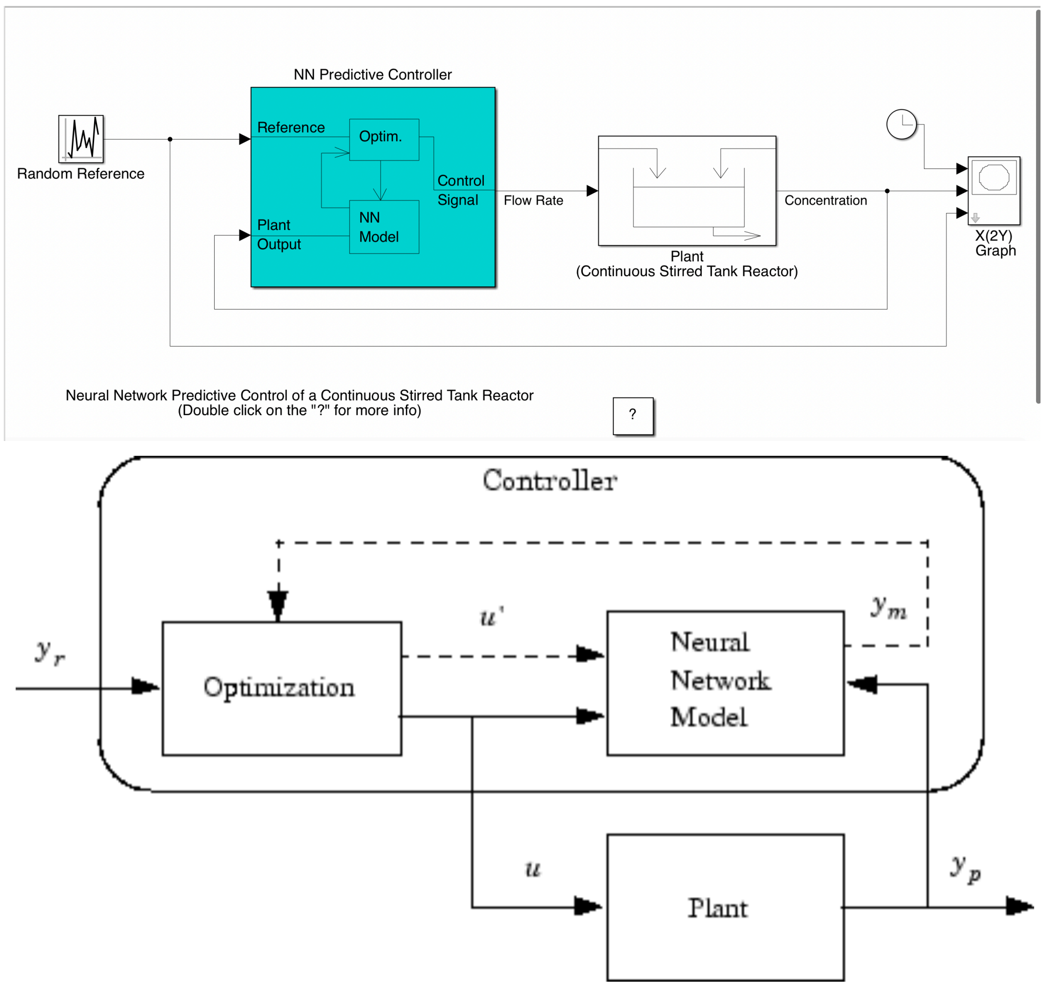

Figure 13: NN predictive controller scheme for CSTR

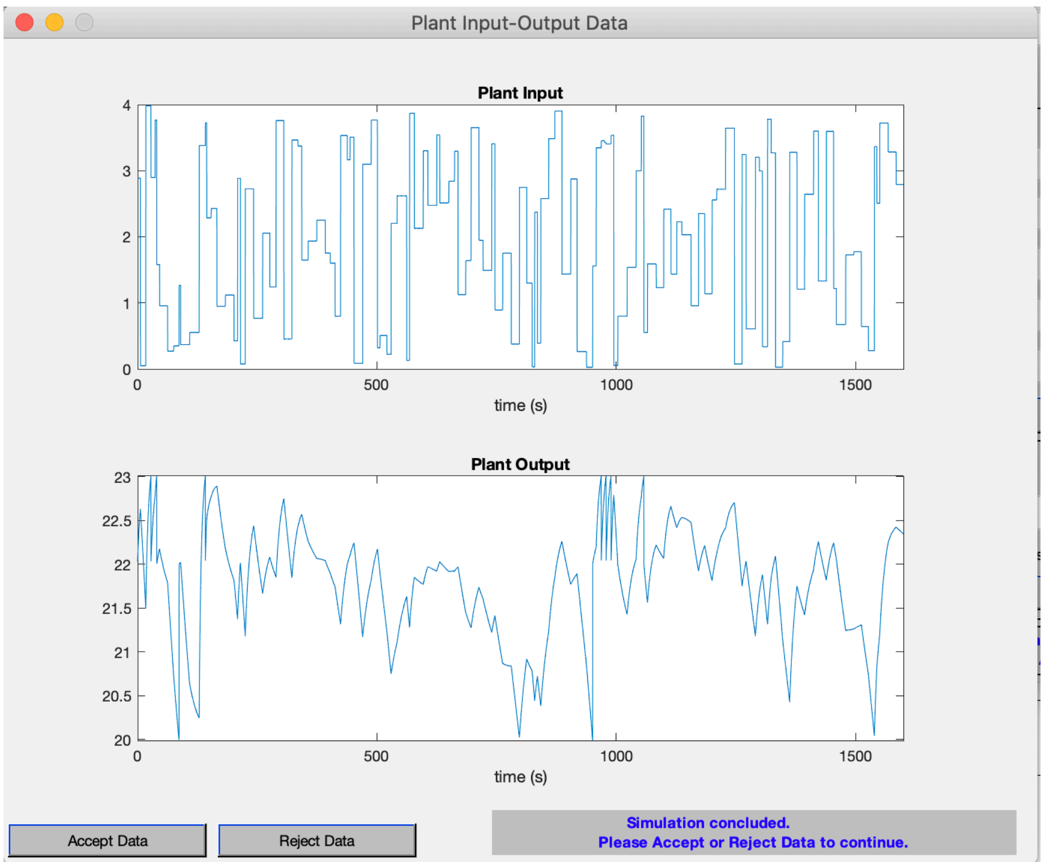

Figure 14: System Identification using input-output data

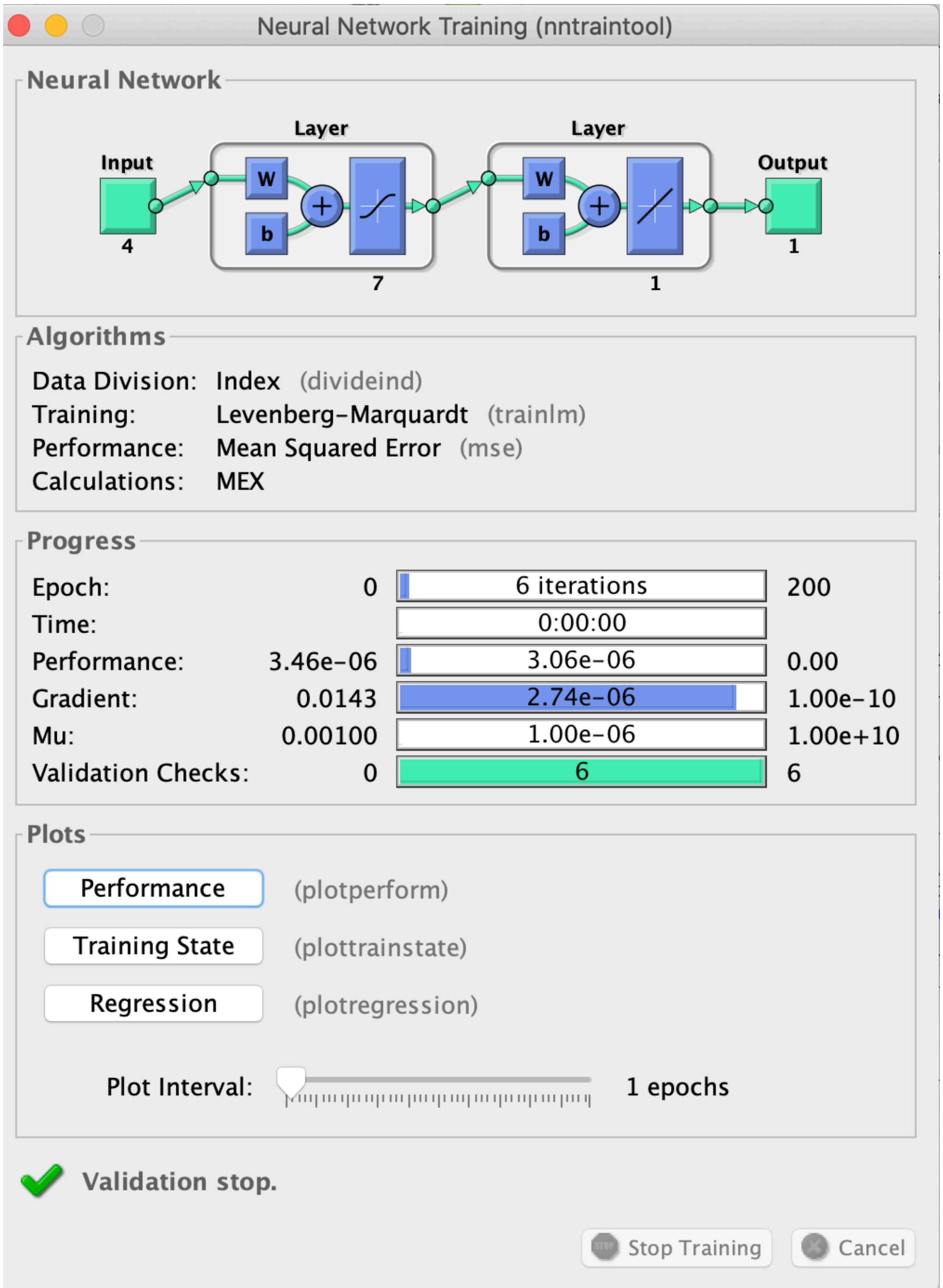

Figure 15: NN Training

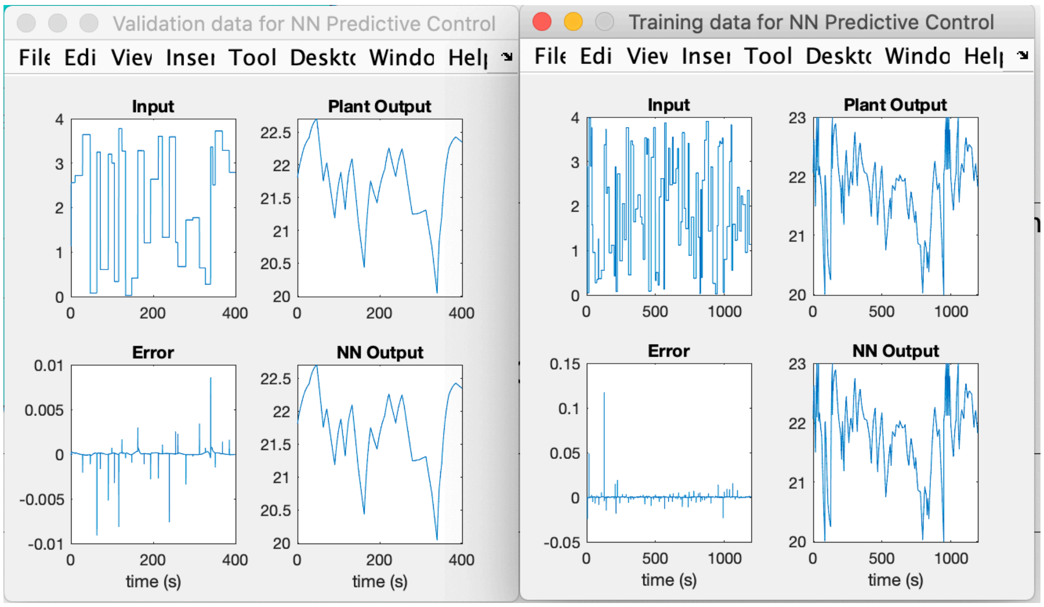

Figure 16: Training error and Validation error

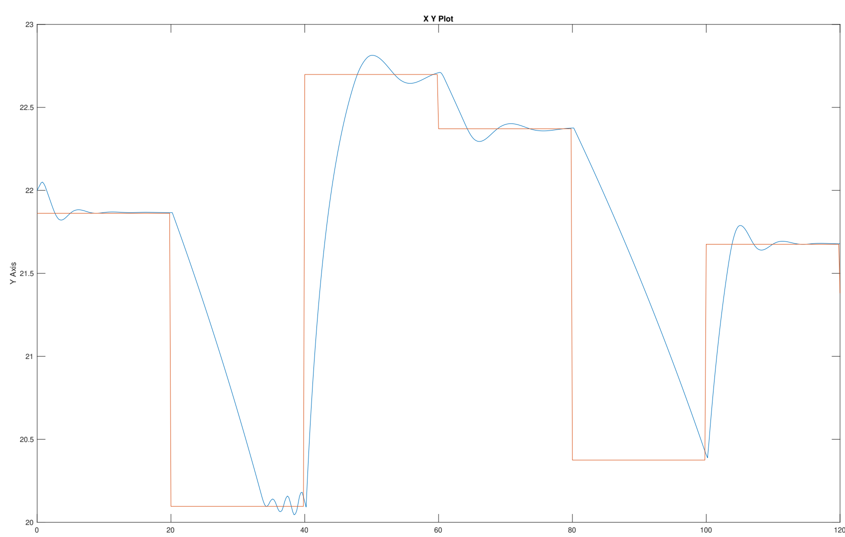

Figure 17: Following reference trajectory using NN predictive controller

5 Discussion and Conclusion

In this report we address the literature and theory behind LBMPC. There are a great deal of opportunities for LBMPC due to the fact that the Machine Learning tools are powerful for data-driven and learning. Based on the structures of LBMPC in which the learning methods take place, the LBMPC approaches can lie into three categories. Firstly, in learning the system dynamics, most researcher have focused on automated improvement of the MPC using learning methods by identifying the true system model and addressing challenges such as fine-tuning for performance during operation, or adjustment of MPC parameters. Secondly, in learning the control design, researchers have tried to increase the performance of MPC using fine-tuning objective function parameters, such as weighting matrices, and designing the largest terminal set to mitigate the conservatism with respect to the robust conventional MPC. Finally, in safe learning, the conventional MPC firstly guarantees the feasibility of the system (called safety filter) and then the learning methods are employed to improve the performance of the closed-loop system. To wrap it up, although the LBMPC approaches can deal with system uncertainty and improve the closed-loop performance of the system, there should be further development in improving computational methods for real-time implementation and also in the field of safety control for some important applications.

Bibliography

[1] A. Aswani, H. Gonzalez, S. S. Sastry, and C. Tomlin, "Provably safe and robust learning-based model predictive control," *Automatica*, vol. 49, pp. 1216–1226, May 2013.

[2] L. Hewing, K. P. Wabersich, M. Menner, and M. N. Zeilinger, "Learning-Based Model Predictive Control: Toward Safe Learning in Control," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, no. 1, pp. 269–296, 2020. _eprint: https://doi.org/10.1146/annurev-control-090419-075625.

[3] U. Rosolia, X. Zhang, and F. Borrelli, "Data-Driven Predictive Control for Autonomous Systems," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, no. 1, pp. 259–286, 2018. _eprint: https://doi.org/10.1146/annurev-control-060117-105215.

[4] A. Mesbah, "Stochastic model predictive control with active uncertainty learning: A Survey on dual control," *Annual Reviews in Control*, vol. 45, pp. 107–117, Jan. 2018.

[5] A. Mesbah, "Stochastic Model Predictive Control: An Overview and Perspectives for Future Research," *IEEE Control Systems Magazine*, vol. 36, pp. 30–44, Dec. 2016. Conference Name: IEEE Control Systems Magazine.

[6] A. Aswani, P. Bouffard, and C. Tomlin, "Extensions of learning-based model predictive control for real-time application to a quadrotor helicopter," in *2012 American Control Conference (ACC)*, pp. 4661–4666, June 2012. ISSN: 2378-5861.

[7] B. Recht, “A Tour of Reinforcement Learning: The View from Continuous Control,” *arXiv:1806.09460 [cs, math, stat]*, Nov. 2018. arXiv: 1806.09460.

[8] T. A. N. Heirung, B. Foss, and B. E. Ydstie, “MPC-based dual control with online experiment design,” *Journal of Process Control*, vol. 32, pp. 64–76, Aug. 2015.

[9] T. Parisini and R. Zoppoli, “A receding-horizon regulator for nonlinear systems and a neural approximation,” *Automatica*, vol. 31, pp. 1443–1451, Oct. 1995.

[10] A. Domahidi, M. N. Zeilinger, M. Morari, and C. N. Jones, “Learning a feasible and stabilizing explicit model predictive control law by robust optimization,” in *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pp. 513–519, Dec. 2011. ISSN: 0743-1546.

[11] S. Lucia and B. Karg, “A deep learning-based approach to robust nonlinear model predictive control,” *IFAC-PapersOnLine*, vol. 51, pp. 511–516, Jan. 2018.

[12] M. Hertneck, J. Köhler, S. Trimpe, and F. Allgöwer, “Learning an Approximate Model Predictive Controller with Guarantees,” *IEEE Control Syst. Lett.*, vol. 2, pp. 543–548, July 2018. arXiv: 1806.04167.

[13] Z. Wu, A. Tran, D. Rincon, and P. D. Christofides, “Machine learning-based predictive control of nonlinear processes. Part I: Theory,” *AIChE Journal*, vol. 65, no. 11, p. e16729, 2019. _eprint: https://aiche.onlinelibrary.wiley.com/doi/pdf/10.1002/aic.16729.

[14] E. Terzi, L. Fagiano, M. Farina, and R. Scattolini, “Learning-based predictive control for linear systems: A unitary approach,” *Automatica*, vol. 108, p. 108473, Oct. 2019.

[15] G. Chowdhary, M. Mühlegg, J. P. How, and F. Holzapfel, “Concurrent Learning Adaptive Model Predictive Control,” in *Advances in Aerospace Guidance, Navigation and Control* (Q. Chu, B. Mulder, D. Choukroun, E.-J. van Kampen, C. de Visser, and G. Looye, eds.), (Berlin, Heidelberg), pp. 29–47, Springer, 2013.

[16] T. A. N. Heirung, B. E. Ydstie, and B. Foss, "Dual adaptive model predictive control," *Automatica*, vol. 80, pp. 340–348, June 2017.

[17] P. Bouffard, A. Aswani, and C. Tomlin, "Learning-based model predictive control on a quadrotor: Onboard implementation and experimental results," in *2012 IEEE International Conference on Robotics and Automation*, pp. 279–284, May 2012. ISSN: 1050-4729.

[18] A. Aswani, P. Bouffard, and C. Tomlin, "Extensions of learning-based model predictive control for real-time application to a quadrotor helicopter," in *2012 American Control Conference (ACC)*, pp. 4661–4666, June 2012. ISSN: 2378-5861.

[19] A. Aswani, N. Master, J. Taneja, D. Culler, and C. Tomlin, "Reducing Transient and Steady State Electricity Consumption in HVAC Using Learning-Based Model-Predictive Control," *Proceedings of the IEEE*, vol. 100, pp. 240–253, Jan. 2012. Conference Name: Proceedings of the IEEE.

[20] M. Milanese and A. Vicino, "Optimal estimation theory for dynamic systems with set membership uncertainty: An overview," *Automatica*, vol. 27, pp. 997–1009, Nov. 1991.

[21] M. Milanese and C. Novara, "Set Membership identification of nonlinear systems," *Automatica*, vol. 40, pp. 957–975, June 2004.

[22] G. Beliakov, "Interpolation of Lipschitz functions," *Journal of Computational and Applied Mathematics*, vol. 196, pp. 20–44, Nov. 2006.

[23] J.-P. Calliess, "Conservative decision-making and inference in uncertain dynamical systems," p. 271.

[24] A. Mesbah, "Stochastic Model Predictive Control: An Overview and Perspectives for Future Research," *IEEE Control Systems Magazine*, vol. 36, pp. 30–44, Dec. 2016. Conference Name: IEEE Control Systems Magazine.

[25] R. Soloperto, M. A. Müller, S. Trimpe, and F. Allgöwer, "Learning-Based Robust Model Predictive Control with State-Dependent Uncertainty," *IFAC-PapersOnLine*, vol. 51, pp. 442–447, Jan. 2018.

[26] K. P. Wabersich and M. N. Zeilinger, “Safe exploration of nonlinear dynamical systems: A predictive safety filter for reinforcement learning,” *arXiv:1812.05506 [cs]*, Apr. 2019. arXiv: 1812.05506.

[27] C. E. Rasmussen and C. K. I. Williams, *Gaussian processes for machine learning*. Adaptive computation and machine learning, Cambridge, Mass: MIT Press, 2006. OCLC: ocm61285753.

[28] M. Neumann-Brosig, A. Marco, D. Schwarzmann, and S. Trimpe, “Data-efficient Auto-tuning with Bayesian Optimization: An Industrial Control Study,” *IEEE Trans. Contr. Syst. Technol.*, vol. 28, pp. 730–740, May 2020. arXiv: 1812.06325.

[29] U. Rosolia and F. Borrelli, “Learning Model Predictive Control for Iterative Tasks. A Data-Driven Control Framework,” *IEEE Transactions on Automatic Control*, vol. 63, pp. 1883–1896, July 2018. Conference Name: IEEE Transactions on Automatic Control.

[30] C. Finn, S. Levine, and P. Abbeel, “Guided Cost Learning: Deep Inverse Optimal Control via Policy Optimization,” *arXiv:1603.00448 [cs]*, May 2016. arXiv: 1603.00448.

[31] D. Seto, B. Krogh, L. Sha, and A. Chutinan, “The Simplex architecture for safe online control system upgrades,” in *Proceedings of the 1998 American Control Conference. ACC (IEEE Cat. No.98CH36207)*, vol. 6, pp. 3504–3508 vol.6, June 1998. ISSN: 0743-1619.