Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Disturbance Rejection in Neural Network Model Predictive Control

Ali Jazayeri *. Alireza Fatehi, **
Houman Sadjadian*, Ali Khaki-Sedigh******

* *Advanced Process Automation and Control (APAC) Research Group, Faculty of Electrical Eng.*
K. N. Toosi University of Technology Tehran, Iran
(sajazayerim@yahoo.com, fatehi@kntu.ac.ir, sadjadian@dena.kntu.ac.ir, sedigh@kntu.ac.ir)

Abstract: Neural Network Model Predictive Control (NN-MPC) combines reliable prediction of neural network with excellent performance of model predictive control using nonlinear Levenberg-Marquardt optimization. It is shown that this structure is prone to steady-state error when external disturbances enter or actual system varies from its model. In this paper, these model uncertainties are taken into account using a disturbance model with iterative learning which adaptively change the learning rate to treat gradual effect of the model mismatch differently from the drastic changes of external disturbance. Then, a high-pass filter on error signal is designed to distinguish disturbances from model mismatches. Practical implementation results as well as simulation results demonstrate good performance of the proposed control method.

1. INTRODUCTION

Model Predictive Control (MPC) is one the most successful approaches both in academia and the industries. An important reason for this success is the concept of prediction in control.
Conventional MPC is based on a linear model of a system (Camacho et al., 2007). However, most actual systems are indeed nonlinear when they work in a wide range of operating points. There are some approaches in which predictive concept is mixed with nonlinear controllers. Two reasons limit conventional MPC to be linear.: Linear system identification and Linear optimization of MPC cost function.
The first aspect that limits linear MPC is linear identification in which extracted mathematical model is linear. A good substitute for linear identification is neural network identification (Nelles, 2001) which is proved to identify a general nonlinear system (Akesson et al., 2005).
The other important factor which justifies the use of a nonlinear model is that analytical solution of linear MPC does not exist in nonlinear case and the optimizer has to numerically minimize the MPC cost function. Different numerical approaches have been proposed for Nonlinear MPC. Soloway et al. (1996) suggests Newton-Raphson to solve the nonlinear optimization and Gil et al. (2000) used gradient descent. Also, Leskens et al. (2005) used sequential quadratic programming while Levenberg-Marquardt (LM) is faster solution than others (Nogaard et al., 2003).
Adaptive concept of the controller requires the neural network identifier to learn in real-time. There are several approaches to train the neural network in real time identification. In Ng (1997) Learning by Recursive Least Square (LRLS) method is discussed to adaptively train neural network. Basically, this method uses linear Recursive Least Square (RLS) concept to train weights of the neural network in each sampling period. Because the number of data at each iteration is not rich, this method usually requires a long time to be initiated or to catch any variations in the identified system. Number of data which are required to train a neural network increases exponentially by the number of training parameters.
In another approach, Moragado et al. (2005) used Back-Propagation-Through-Time with sliding window (Haykin, 1999). In this paper, this latter method is chosen to train the network efficiently while keeping the identification up-to-date and being able to keep track of variation of the system. However, since network training requires rather huge amount of computations, this training is done in batch. In other words, the real computational aspect results in some periods where the network does not train. As a consequence, taking account that the controller is model-based, model mismatch or disturbance may cause low frequency or steady-state errors. In this paper, a model-based problem is resolved by adding a new disturbance model to identify disturbances iteratively.
In the nonlinear case, Akesson et al. (2005) suggests to directly subtract approximated prediction error from neural network output to obtain offset-free identifier. Another way is to apply an outer loop integrator controller (Wang et al., 1998). Both approaches are fairly simple but affect normal operation of control system greatly. Alternatively, Kuure-Kinsey et al. (2006) used Kalman filter to avoid steady-state offset. It is also possible to put a pre-filter in reference signal to change it suitably (Gil et al., 2000). This feedforward pre-filter is itself a model-based system and depends on the operating point of the system.
In this paper, a new disturbance identifier is developed to efficiently estimate the external disturbances and model mismatches to eliminate the steady-state error of constrained nonlinear model predictive control. Neural network training is done using the gradient descent method for data of a sliding window. Disturbance model is trained by a gradient descent method with adaptive weighting that distinguished external disturbances and model mismatches.

2. NEURAL NETWORK IDENTIFICATION

The neural network (NN) model of the system predicts output of the system to use in the cost function of model predictive control. It relates an implicit relationship between control input (*u*) and system output (*y*). Using this NN model, future outputs of objective function are represented by future control signals.

Neural network model of the system is a Multilayer Perceptron (MLP) with one output neuron because the system is assumed single-input single output. Delayed control signals and control outputs enter the network in predictor structure (Nelles, 2001):

$$\hat{y}(k) = F(x(k)) \tag{1}$$

Where, F is the neural network function and

$$x = \begin{bmatrix} y(k-1) \\ \vdots \\ y(k-n_a) \\ u(k-d) \\ \vdots \\ u(k-d-n_b+1) \end{bmatrix} \tag{2}$$

Where, d is input delay of the system, n_a and n_b are number of delayed outputs and inputs of the network. Output of the network is the one step-ahead prediction of the system. It can predict $\hat{y}(t+1)$ but to predict $\hat{y}(t+2)$ to $\hat{y}(t+N_2)$, the neural network feed predicted output as its input.

3. CONSTRAINED OPTIMIZATION OF THE MPC COST FUNCTION

The model described in previous section predicts future outputs of the system. To have a model predictive control, cost function (3) should be minimized:

$$J_1 = \sum_{k=N_1}^{N_2} (\hat{y}(t+k) - r(t+k))^2 + \sum_{k=1}^{N_u} (\lambda_k \Delta u(t+k))^2 \tag{3}$$

The first term in (3) is future error and the second term is variation of control signal. r is the reference or tracking signals. u is the control signal and Δu is the difference of the control signal. λ_k is the weight factor of *k*-th future control signal which determines importance of control signal in minimization. Larger values of λ will cause smoother control signals. N_1 , N_2 and N_u are the minimum output horizon, maximum output horizon and control horizon respectively. Larger control horizon normally improves performance at the cost of more computational burden. Minimization obtains best control signals over control horizon but within these best control signals only first one applies to the system and others are just deleted. This process repeats each iteration.

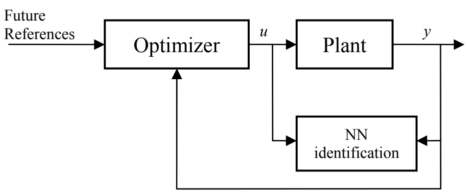


Fig. 1 –Block Diagram of Neural Network Model Predictive Control

As mentioned, the cost function (3) depends on the predicted outputs which can be presented as a function of the previous output signals and the future control signals. To minimize the cost function (3) different methods were described in section 1. In linear MPC, analytical solution exists (Camacho, 2007) and its stability is proven. However, in nonlinear case, minimization is a numerical problem with different approaches and very limited stability analysis. One good example of stability analysis of NN-MPC is Huang (2003) where the local linear model of the plant is considered. To solve the minimization problem, one of the most efficient approaches is Levenberg-Marquardt for both high precision and low computational process time. In the next section, constraint on control signals is considered in the cost function (3).

3.1 Constraint on control signal

The cost function (3) is minimized by changing future control signals. In all control applications there are inequity constraint on control signals. For example, control signals such as thrust of an aircraft or opening of a valve is inevitably limited between certain physical bounds. Some linear MPC's such as MAC and DMC intrinsically considers the constraints in contrast to GPC (Camacho, 2007). In nonlinear MPC where minimization is done iteratively, a term is usually added to the cost function to consider constraints. This term is constant in working area and increases when control signal (constrained state) approaches the bounds. This increase must be rapid but continuous (Soloway et al., 1996) because its second derivative must exist for L-M method.

In this paper, the term is an exponential function with a sharpness variable *S*:

$$f(u) = \exp\left(\frac{2S.(u-\bar{u})sign(u-\bar{u})}{u_{\max} - u_{\min}}\right) / \exp(S) \tag{4}$$

Where, f is the constraint function that is defined to increase near limits (Fig. 2). Variable S is the sharpness of the function *f*. $u_{\min}$ and $u_{\max}$ are lower and upper bounds of control signal, respectively. Also, $\bar{u}$ is the average of lower and upper bounds. Fig.2 illustrates the effect of sharpness variable on shape of the function *f*. It can be seen that when sharpness increases, the cost function approaches the ideal cost function. From now on in this paper, sharpness is assumed to be 100 in all cases.

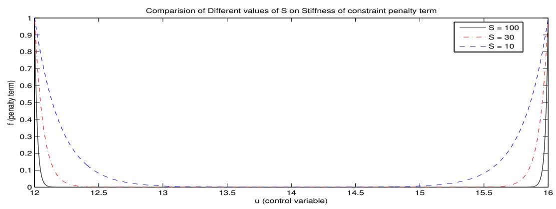


Fig.2 – Constraint function with different sharpness values

The obtained cost function will be added to the main cost function of MPC (3) to form a cost function with constraint.

Equation (4) is just for one control signal while there are N_u control signals in the cost function (3). Total constraint term is will is:

$$J_2 = \sum_{k=N_1}^{N_2} (\hat{y}(t+k) - r(t+k))^2 + \sum_{k=1}^{N_u} (\lambda_k \Delta u(t+k))^2 + \sum_{k=1}^{N_u} f(u(t+k)) \tag{5}$$

To find minimum of Eq. 6 using numerical approaches like L-M, it is necessary to calculate Gradient and Hessian of the cost function (5).

Using neural network model of the system, it is straightforward to calculate Gradient and Hessian of the cost function (Nogaard et al., 2003).

Search direction is now obtained using Levenberg-Marquardt (L-M) search direction as follows

$$U_{k+1} = U_k + \mu_k f_k \quad (6)$$

Where, U_k is the control signal vector, μ_k is the step size, f_k is the search direction and all of them are in *k*-th iteration. L-M method suggests (7) to find new search direction.

$$\left(H[U_k(t)] + \lambda_k I\right) f_k = -G[U_k(t)] \quad (7)$$

Where, *H* is the Hessian and *G* is the Gradient of the cost function, *I* is the identity matrix and λ_k is the Levenberg-Marquardt parameter. When λ_k is small, this method is a quadratic approximation and when it is large and Hessian is negligible, L-M method works like simple gradient method. At first iterations, L-M works as a gradient method and as it gets near the optimal point it gradually switches to Newton-based methods. When L-M parameter gets smaller, L-M finds a locally linear solution but precisely and quickly. After each iteration of the search, Hessian is checked to be positive definite (convex optimization). If Hessian is not positive definite, λ_k is increased until this happens. To investigate positive definiteness of Hessian Cholesky factorization is used (Nogaard et al., 2003).

4. DISTURBANCE MODELLING

The controller which is described in the previous section does not train for a while. Between two training stages, control system is model-based and external disturbances or model mismatches cause steady state error. In addition, training of the identifier network is a time-consuming process and cannot run at each iteration. Still a good controller should detect and reject disturbances in the same iteration as they come.

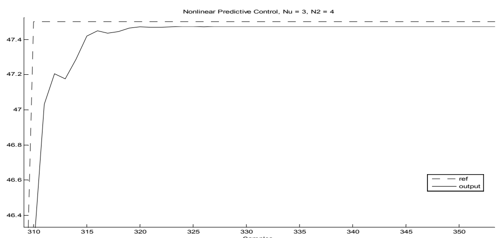


Fig 3. Model mismatch effect in simulation of NNMPC

This is a serious problem in model-based control systems. If the neural network model differs from actual system, controller output does not converge to the desired reference. Fig. 3 shows the effect of misleading prediction on control performance of the NN-MPC in simulation.

As illustrated in Fig. 3, there is a steady state error between output signal and the reference signal.

4.1 Structure of Disturbance Model

The disturbance model should take the advantages of RLS. It should have simple structure with few adaptive parameters and fast learning algorithm working parallel to the main neural network identifier. In spite of the main neural network, disturbance model should be updated in each sampling time.

Proposed disturbance model adds to the main neural network to modify the predicted output (1) to

$$\hat{y}(k) = F(x(k)) + d_M(k) \quad (8)$$

Where, the output of the disturbance model. is

$$d_M(k) = w_d \hat{e}(t) + b \quad (9)$$

Where, $\hat{e}(t)$ is difference between the main neural network output and the actual system output. b and w_d are disturbance model weightings which adapt in every sampling period with a modified gradient descent method. Gradient descent proposes the following adaptation rule:

$$b(k+1) = b(k) + \eta \hat{e}(k) \quad (10)$$

Where, η is the learning rate. Equation (10) is equivalent to:

$$\Delta b = \eta \hat{e} \quad (11)$$

which can be rewritten as:

$$b = \frac{1}{\Delta} \eta \hat{e} \quad (12)$$

Equation (12) is just like an integrator with particular effects of an integrator controller such as long settling time or large overshoot. To improve the performance, it is proposed to apply a Proportional-Integral learning rule:

$$b = \frac{1}{\Delta} \eta \hat{e} + k_p \hat{e} \quad (13)$$

Where, η and k_p are the integral and proportional coefficients, respectively. Equation (13) can be simplified as:

$$\Delta b = \eta \hat{e} + k_p \Delta \hat{e} \quad (14)$$

Summarizing the rules to adapt the disturbance model parameters, we obtain:

$$\begin{aligned} b_k &= b_{kt-1} + \eta \hat{e}_k + k_p\left(\hat{e}_k - \hat{e}_{k-1}\right) \\ w_d(k+1) &= w_d(k) + \eta \hat{e}^2(t) \end{aligned} \quad (15)$$

Equation (15) adapts much faster than (10).

Disturbance modelling deals with two kinds of problems. First, it compensates external unknown disturbances. Second, it reduces the effect of gradual model mismatches. Both cause prediction error but they are caused by different sources and should be treated differently. For external disturbances, learning rates should be larger.

In this paper we distinguish disturbances from model mismatch and apply different weightings. This is called adaptive weightings of disturbance modeling. Disturbance needs faster adaptation to catch up with rapid changes of the system. When there is no disturbance, adaptation should be slowed down because higher adaptation may decrease model predictive transient performance.

To distinguish the disturbance it is possible to use approach similar to what (Hägglund et al., 2000) proposed and to use a high-pass filter on error signal with pole near time constant of the system. The error is defined to be the difference between the main neural network output and the actual system output. Disturbance model has no effect to this error. The main advantage of defining error without interference of disturbance model is that this error is able to detect disturbances simultaneous with set-point changes. Additionally, it detects non step-shaped disturbances. A potential problem in this approach might happen after a long time when the error between neural network and actual output accumulates. To avoid long time problems, difference of the error is considered. It is equivalent to put a high-pass filter with large pole over the error signal. Therefore, the digital high-pass filter is:

$$G_F(z) = \frac{(1-\omega_{hp}) - z^{-1}e^{-\omega_{hp}T}}{1-z^{-1}e^{-\omega_{hp}T}} \qquad (16)$$

Where, $\omega_{hp} = \frac{1}{T}$ and T is time constant of the system. In digital filter (16) when ω_{hp} is large, the zero is actually negligible and the high-pass filter is a difference function asymptotically. When output of this filter exceeds its threshold, there is a disturbance.

After disturbance detection, training of disturbance model is accentuated (adaptive learning rate for disturbance model). This fast learning should remain until disturbance is rejected completely. As proposed in (Hägglund et al., 2000) the disturbance rejection should work for at least one time constant of the system.

In summary, detecting external disturbances by a high-pass filter has two advantageous features over conventional disturbance detection. First, this approach detects disturbances of any shape (it does not limit identified disturbances to step-shaped). Second, disturbance can be detected even if it occurs exactly when set point changes.

5. SIMULATION RESULTS

In this section disturbance rejection of the proposed model predictive controller verified through computer simulation.

In Fig. 4, an external step-shaped disturbance enters at t=2000 second in linear MPC. When disturbance occurs, output starts fluctuating and control signal is not smooth. However, after around 500 seconds, the linear MPC damps the effect of the disturbance and the overall performance become reasonable.

For nonlinear MPC, three cases were considered in the simulation. In the first case, NN-MPC works without any disturbance model or disturbance rejection scheme. External disturbance in this case causes a bias in output from the desired value (Fig. 5).

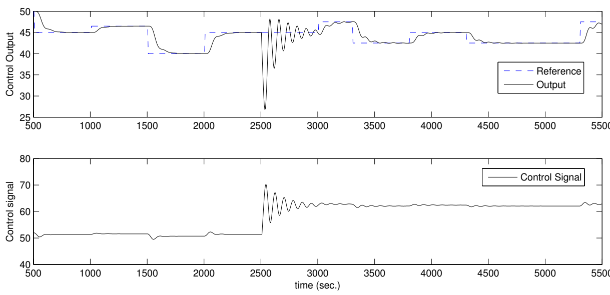


Fig. 4 – Performance of the linear MPC when an unknown unmodeled disturbance enters

In the second nonlinear simulation, disturbance rejection is done by a disturbance model with constant learning parameters which is slow in rejection of step-shaped external disturbances (Fig. 6).

In the final simulation for nonlinear MPC, disturbance model distinguishes external disturbances from model mismatches and adaptively changes learning parameters to reject external disturbances. As shown in Fig. 7, this approach is the fastest one in disturbance rejection.

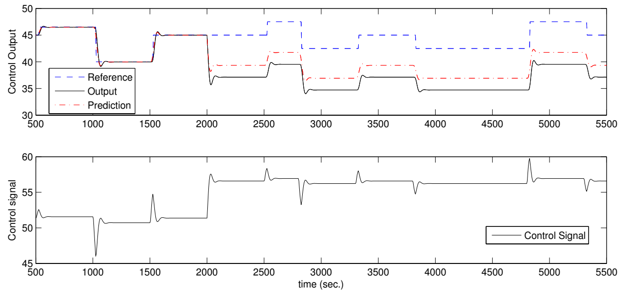


Fig. 5 – Performance of NNMPC when an unknown unmodeled disturbance enters without any disturbance rejection scheme

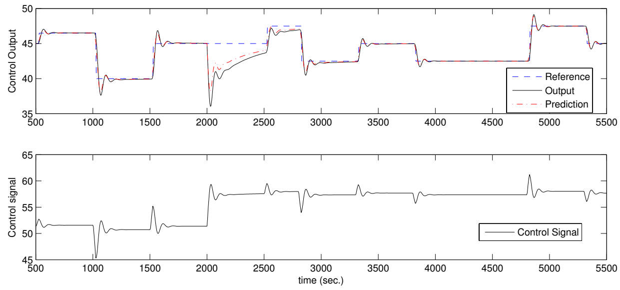


Fig. 6 – Performance of NNMPC when an unknown unmodeled disturbance enters with constant disturbance rejection

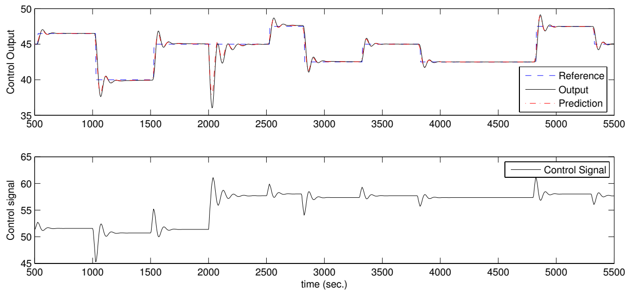


Fig. 7 – Performance of NNMPC when an unknown unmodeled disturbance enters with adaptive disturbance rejection

It is important to notice that the entered disturbance has made a drastic change in system parameters, which can be comprehended by noticing the control signal in Fig. 7.

6. CASE STUDY – NONLINEAR WATER LEVEL PLANT

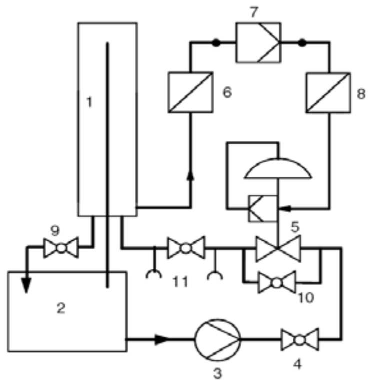
To verify applicability, the proposed method is applied to control a nonlinear water level plant. In this section, first physical modelling of the system is introduced and then some practical and implementation problems are pointed out. Finally, practical results of the proposed method are compared with linear MPC both with and without disturbances in different operating points.

6.1 Physical Modelling

The proposed method was implemented on a lab-scale water tank system RT512 made by GUNT company. Fig. 8 and Fig. 9 show the plant and its P&ID respectively.

It is desired to control the water level in tank (1). Hand valve (8) determines flow of outlet water from the bottom of the main tank. A big reservoir tank (2) gathers outlet water and a pump (3) circulates the water. Flow of the pumped water is controlled by a control valve (5). This water pours to the main tank. A Level sensor (5) measures the level of water in the main tank. Controller (6) should observe this measurement and apply appropriate command to the control valve.

Fig.8 – Water Level Plant, Process Control Lab. – Electrical Engineering Department, K. N. Toosi University of Technology



1 Level Tank
2 Reservoir
3 Pump
4 Inlet Cock
5 Control Valve
6 Pressure Transducer
7 External Controller
8 Current/Pressure Transducer
9 Outlet Cock
10 Bypass
11 Hose connection for cascade operation

Fig. 9 – P&ID of Water Level Plant

In actual implementation it is important to note that choosing the sampling time and the corresponding horizons is a trade-off between performance and computational burden. Another point is that identification data should be persistent excited (P.E.) that normally does not satisfy in tight control when all of the signals are constant.

5.2 Practical Results

The proposed approach is implemented on the nonlinear level plant and compared to a typical conventional MPC in an identical situation. Prediction horizon, sampling time, reference trajectory and disturbance size are the same.

The first experiment examines features of linear MPC. As shown in Fig. 10, MPC method tracks desired trajectory without steady state error. However, external disturbance (changing hand valve from 45 to 55 at t = 1800 sec.) requires a long time to be compensated (Fig. 11). This disturbance even shifted the system output to saturation.

The second experiment shows the performance of the nonlinear MPC. Without disturbance modelling, there is a steady state error as demonstrated in Fig. 12. This error becomes larger when an external disturbance enters at t = 3500 sec. As a result of the disturbance in the model-based system, output moved to a biased point far from the desired trajectory.

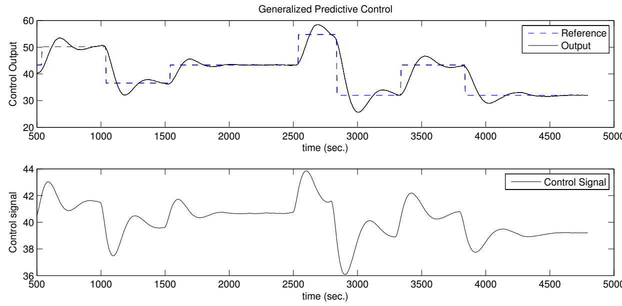


Fig. 10 – Tracking result in practical experiment, linear MPC

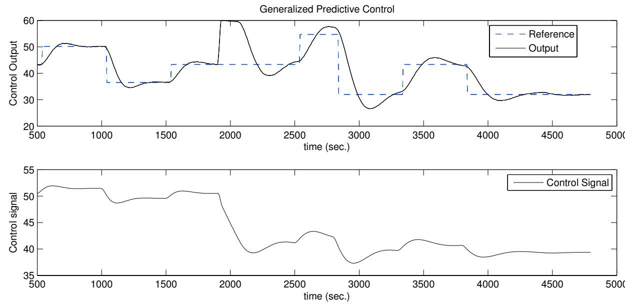


Fig. 11 – Disturbance rejection, linear MPC

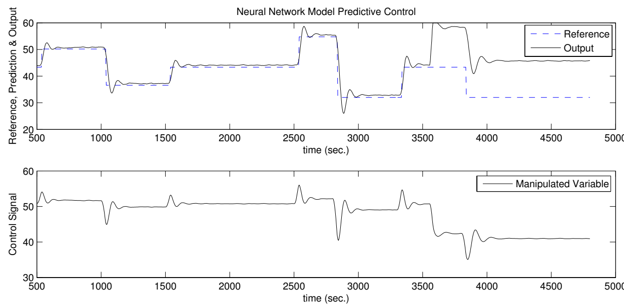


Fig. 12 – Tracking result in practical experiment, Model-Based NN-MPC without disturbance modelling

The final experiment is on nonlinear MPC with disturbance model and adaptive learning parameters (Fig. 13). There is no steady state error in tracking of the desired reference.

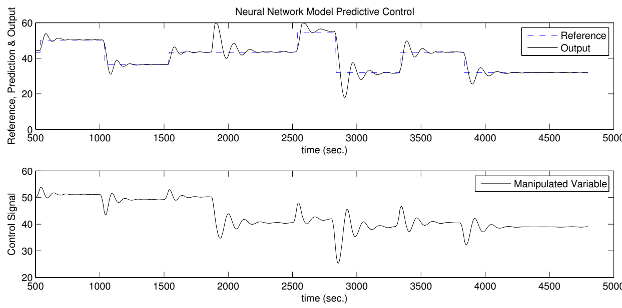


Fig. 13 – Tracking result in practical experiment, NN-MPC with disturbance modelling and adaptive learning parameters

Comparing linear and nonlinear MPC (Fig. 11 and Fig. 13, respectively), it is obvious that disturbance rejection occurs faster in the nonlinear case.

6. CONCLUSIONS

In this paper, a neural network is used for system identification. Sliding window with time-to-time learning algorithm is done. Model uncertainties and mismatches as well as external disturbances were identified using an online easy-to-learn, small disturbance network. This network adapts to disturbances more quickly than to model mismatches. To distinguish disturbances from model mismatches, a high-pass filer of error signal is considered. Both computer simulation and practical implementation on a lab-scale nonlinear water level plant demonstrate better performance of the proposed adaptive disturbance modeling in comparison to NN-MPC with and without constant disturbance modeling and also linear MPC.

REFERENCES

Akesson B. M. , Toivonen H. T. , Waller J. B. and Nystrom R. H. (2005) 'Neural network approximation of a nonlinear model predictive controller applied to a pH neutralization process' *Journal of Computers & Chemical Engineering*, Volume 29, Issue 2, pp. 323-335

Camacho E. F. and Bordons C. (2007) *Model Predictive Control,* Springer

Moragado Dias F. , Antunesa A., Vieirab J., and Manuel Mota A. (2005) 'On-line Training of Neural Networks: A Sliding Window Approach for the Levenberg-Marquardt Algorithm' *Artificial Intelligence and Knowledge Engineering Applications: A Bioinspired Approach*, Springer, pp. 577-585

Gil P. , Henriques J. , Dourado A and Duarte-Ramos H. (2000) 'Extended Neural Model Predictive Control of Non-linear Systems', *Proceeding of . IASTED - ASC2000*, pp. 94-100

Hägglund T. and Astrom J. K. (2000) 'Supervision of Adaptive Control Algorithms' *Automatica*, Volume 36, Number 8, pp. 1171-1180

Haykin S. (1999) *Neural Networks: A Comprehensive Foundation,* Prentice-Hall

Jalali J. and Ghale A. P. (2005) 'Self-Tuning Regulators for Water Level Plant', Bachelor's Thesis in Electrical Engineering-control, K. N. Toosi University of Technology

Kuure-Kinsey M. , Cutright R. and Bequette B. W. (2006) 'Computationally Efficient Neural Predictive Control based on a Feedforward Architecture' *Industrial and Engineering Chemistry Research* , Volume 45 , pp. 8575-8582

Leskens M. , Kessel J. L. F. van, Hof P. M. J. van den and Bosgra O. H. (2005) 'Nonlinear Model Predictive Control with Moving Horizon State and Disturbance Estimation – with Application to MSW Combustion' *Proceedings of 16th IFAC World Congress,* pp. 1-6

Nelles O. (2001) *Nonlinear System Identification: From Classical Approaches to Neural Network and Fuzzy Models,* Springer

Ng G. W. (1997) *Application of Neural Network to Adaptive Control of Nonlinear Systems,* John Wiley

Nogaard M. , Ravn O. , Poulsen N. K. and Hansen L. K. (2003) *Neural Networks for Modelling and Control of Dynamic Systems*, Springer-Verlag

Soloway D. and Haley P. J. (1996) 'Neural Generalized Predictive Control; A Newton-Raphson Implementation' *Proceeding of IEEE International Symposium on Intelligent Control,* pp.277–282

Wang H and Wu J. (1998) 'Eliminating the DC component in steady state tracking error for unknown nonlinear systems: a combination of fuzzy logic and a PI outer loop' *Proceedings of the American Control Conference,* Volume 3, pp. 1415-1416