2018 European Control Conference (ECC)
June 12-15, 2018. Limassol, Cyprus

Event-based Observer and MPC with Disturbance Attenuation using ERM Learning

Jaehyun Yoo, Ehsan Nekouei and Karl H. Johansson

***Abstract*— This paper presents a learning-based approach for disturbance attenuation for a non-linear dynamical system with event-based observer and model predictive control (MPC). Using the empirical risk minimization (ERM) method, we can obtain a learning error bound which is function of the number of samples, learning parameters, and model complexity. It enables us to analyze the closed-loop stability in terms of the learning property, where the state estimation error by the ERM learning is guaranteed to be bounded. Simulation results underline the learning's capability, the control performance and the event-triggering efficiency in comparison to the conventional event-triggered control scheme.**

I. INTRODUCTION

Event-based control as a sampled data control scheme has been developed aiming at reducing the communication in networked control systems, by sending information at the observer side only when the desired event condition cannot be guaranteed. Accordingly, the event-based control, represented by model predictive control (MPC) in this paper, saves computational energy as well, by executing the control update when either the information is given from the observer or the prediction horizon is expired. However, if there is large disturbance not to be compensated, the event-triggering instants occur too frequently to make the event-based control scheme meaningless [1], [2].

When disturbance comes from model uncertainty, which is dependent to system state and control variables, there have been a variety of studies to handle the unknown uncertainty, such as supervised learning, system identification, and adaptive control techniques. This paper proposes a supervised learning technique for finding an estimator for disturbance attenuation. The proposed estimator operates based on the training data samples which contain information regarding the inherent relationship between the true uncertainty and the variables of dynamical states and control inputs. The system identification [3] can have the similar purpose to estimate a model from observation data, but it focuses on establishing convergence of the estimation error when the number of samples tends to infinity. When the objective is to analyze the control performance with respect to learning performance for a given a fixed number of samples, the supervised learning methods are more suitable than the system identification technique.

Some adaptive control techniques applying structure of neural network [4], [5] have been reported to deal with unknown disturbance. Rather than using training data, they update the parameters of the neural network based on Lyapunov theory to guarantee stability. However, these methods do not usually account for the randomness of the disturbance.

Many supervised learning [6]–[10] have been developed for disturbance attenuation problem. However, they do rarely address performance guarantees in control theory perspective with respect to learning influences. The analysis of the learning effects on the control system can help a control system designer to figure out how many the number training samples, how much the learning model complexity, and what range of learning parameters are required to acheive a given control object. These are important concepts in designing and assessing the learning-based control system for realistic applications.

The main contribution of this paper is applying a supervised learning technique to compensate for the disturbance and to improve the triggering efficiency and the control performance, while providing stability analysis with respect to the learning properties. We employ empirical risk minimization (ERM) method for learning. ERM is a fundamental statistical learning theory, which is the basis of rich learning techniques such as neural network, logistic regression, and support vector machine [11]. Using the ERM method, we can obtain an upper-bound on the learning error. The bound depends on the variables of the model complexity, the learning parameters, and the number of samples. This learning error bound not only depends on the state estimation error, but also depends on an upper bound of stationary state. The proposed algorithm is numerically evaluated and its efficiency is shown via simulation.

The rest of this paper is organized as follows. Section II presents the system description. Section III suggests the learning-based observer for event-based MPC. Section IV describes the detail of empirical risk minimization. Section V shows the simulation results and concluding remarks are given in Section VI.

II. PROBLEM FORMULATION

Consider the general model of a discrete-time nonlinear system

$$x_{k+1} = f(x_k, u_k) \tag{1}$$

$$y_k = Cx_k, \tag{2}$$

with state $x \in \mathbb{R}^n$ control input $u \in \mathbb{R}^m$, and output $y \in \mathbb{R}^m$, and $f(\cdot, \cdot)$ is a vector-valued unknown nonlinear function. The objective of an observer is to estimate the state x in the presence of unknown function $f(x, u)$ in the system dynamics

Jaehyun Yoo, Ehsan Nekouei and Karl H. Johansson are with the ACCESS Linnaeus Center and the School of Electrical Engineering, KTH Royal Institute of Technology, Stockholm, Sweden. Emails: {jaehyun@kth.se, nekouei@kth.se, kallej@kth.se}

(1). It is assumed that we can select a Hurwitz matrix A such that the pair (A, C) is observable, and similarly we can find B such that (A, B) is controllable, then we can define the system model in the following:

$$x_{k+1} = Ax_k + Bu_k + g(x_k, u_k) \tag{3}$$
$$y_k = Cx_k, \tag{4}$$

where $g(x, u) = f(x, u) - (Ax + Bu) \in \mathbb{R}^n$ that can be treated as the model uncertainty, external disturbance, or matched-input disturbance.

The observer model is defined as follows:

$$\hat{x}_{k+1} = A\hat{x}_k + Bu_k + \hat{g}(\hat{x}_k, u_k) + G(y_k - \hat{y}_k) \tag{5}$$
$$\hat{y}_k = C\hat{x}_k, \tag{6}$$

where $\hat{x}_k$ is the estimated state, and the observer gain $G \in \mathbb{R}^{n\times m}$ can be selected such that $A - GC$ becomes a Hurwitz matrix. It is assumed that the state, input, and disturbance belong to the compact sets:

$$x \in \mathcal{X}, \quad u \in \mathcal{U}, \quad g(\cdot, \cdot) \in \mathcal{G}, \tag{7}$$

where $\mathcal{G}$ is a set of constraint.

For a control mechanism, MPC is considered in this paper. It calculates predictions of current and future control inputs by solving a online finite horizon optimal control problem. The current and predictive states and control inputs are denoted in vector format as

$$\hat{X}(k) = \{\hat{x}(k+i|k)\}_{i=0}^{i=N}, \ \hat{U}(k) = \{\hat{u}(k+i|k)\}_{i=0}^{i=N-1},$$

where N is length of horizon and $x(k|k) = x_k$. Thus, the optimization problem is formulated as follows

$$\min_{\mathbf{u}(k)} J(\hat{X}(k), \hat{U}(k)) = \|\hat{x}(k+N|k)\|_{Q_N}^2 + \sum_{i=0}^{N-1} \left(\|\hat{x}(k+i|k)\|_Q^2 + \|\hat{u}(k+i|k)\|_R^2\right) \tag{8}$$

subject to

$$\begin{aligned} \hat{x}(k+j+1|k) &= A\hat{x}(k+j|k) + B\hat{u}(k+j|k) \\ &\quad + \hat{g}(\hat{x}(k+j|k), \hat{u}(k+j|k)), \\ \hat{u}(k+j|k) &\in \mathcal{U}, \\ \hat{x}(k+j|k) &\in \mathcal{X}, \\ \forall j &= 0, \dots, N-1. \end{aligned} \tag{9}$$

The weighting matrices Q_N, Q, and R are design parameters. The initial state estimation is given by observer:

$$\begin{aligned} \hat{x}(k|k) &= A\hat{x}(k-1|k_\ell) + B\hat{u}(k-1|k_\ell) \\ &\quad + \hat{g}(\hat{x}(k-1|k_\ell), \hat{u}(k-1|k_\ell)) \\ &\quad + G(y(k) - C\hat{x}(k-1|k_\ell)), \end{aligned} \tag{10}$$

where the estimation $\hat{x}(k-1|k_\ell)$ and the optimal control input $\hat{u}(k-1|k_\ell)$ were obtained from the previous event-triggering instant k_ℓ, which have been kept in memory. In time-triggering MPC, it is decided that $\ell = k - 1$. In event-triggering MPC, according to a triggering rule, ℓ is selected among $k - 1 \leq \ell \leq k + N - 2$.

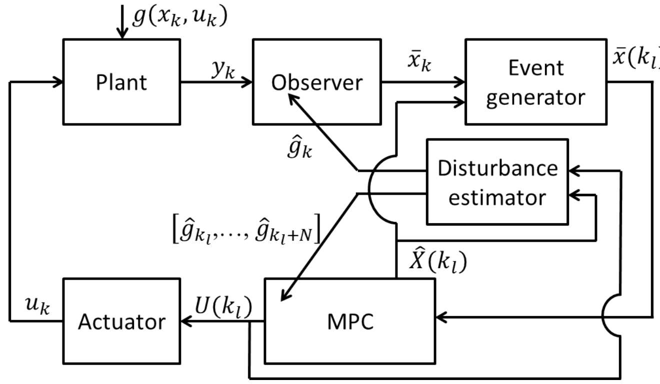


Fig. 1: System architecture of event-based observer and MPC loop. The disturbance estimator is designed by a machine learning technique.

The main purpose of adopting machine learning technique is to model $\hat{g}(x, u)$ in order to identify the nonlinearity of system model and help to estimate the state. First, training samples are collected from repetitive control implements. The obtained training sample set $D = \{(x_{k_i}, u_{k_i})^T, x_{k_i+1}\}_{i=1}^N$, where k_i is a discrete time index in the training phase and $x_{k_i+1} = Ax_{k_i} + Bu_{k_i} + w$ with random noise w. These samples are used to learn the model $\hat{g}(x, u)$. We note that the estimator is desgined in the batch mode thorugh a machine learning technique, so the online learning or the active learning is not considered in this paper.

III. Event-based Observer and MPC

The control architecture for the event-triggered MPC is depiced in Fig. 1. It consists of the plant, the observer, the event generator, the MPC as controller, the actuator choosing one control input among the predictive inputs provided by MPC, and the disturbance estimator. By the event generator, information is sent over the feedback link only if the event condition is satisfied. Between two event-triggered instants or the lapse of the horizon time, the rest of the predictive control inputs are applied to the plant. The discrete-time instants at which the event occurs are denoted by k_ℓ, where $l \in \mathbb{N}$ is the event counter. We set the first event $\ell = 0$ at time $k_0 = 0$. The disturbance estimator is established in a training phase as batch mode.

Without considering observer in event-triggered MPC, the triggering condition usually is based on on state error under the assumption that initial state is given. In case that the initial state must be estimated via an observer, the estimation error is propagated to the predictive state estimates. This paper sets the event-triggering condition by the error of observer-based estimation and state transition estimation originated from the last event, as in [12], [13]. Suppose that we obtained the optimal solution at the last event-triggering instant k_ℓ, given by

$$U(k_\ell) = [u(k_\ell|k_\ell), u(k_\ell+1|k_\ell), \ldots, u(k_\ell+N-1|k_\ell)]. \tag{11}$$

Given the time steps $k_\ell + j$ with $1 \leq j < N$ as the next potential event-triggering instant, we can define two state estimates such that $\hat{x}(k_\ell+j|k_\ell)$ is the estimation propagated from the last event instant k_ℓ, and $\bar{x}(k_\ell+j)$ is the estimation by using a measurement.

The event occurs when either the difference between the two estimates exceeds a certain threshold, or the prediction step exceeds the horizon N, given by:

$$\|\bar{x}(k_\ell+j) - \hat{x}(k_\ell+j|k_\ell)\| \geq \|GC\| \cdot e_{\mathrm{trg}}, \tag{12}$$

$$\text{or } j \geq N. \tag{13}$$

Theorem 1: If the event condition in (12) holds, the expectation of the prediction error $\hat{e}_{k_\ell+j} = x(k_\ell+j) - \hat{x}(k_\ell+j|k_\ell)$ for $0 \leq j \leq N-1$ is bounded as

$$\mathbb{E}[\|\hat{e}_{k_\ell+j}\|] \leq e_{\max}, \tag{14}$$

with

$$e_{\max} = \|A\| \cdot e_{\mathrm{trg}} + e_g, \tag{15}$$

where e_g is the upper bound of $\mathbb{E}\|g_{k_\ell+j-1} - \hat{g}_{k_\ell+j-1}\|$ and $\hat{g}(\equiv \hat{g}(D))$ is designed by an offlie learning based on the training data set D, which will be specified in Section IV.

Proof:

The evolution of the prediction error is in the following.

$$\begin{aligned}
&\hat{e}_{k_\ell+j} = x_{k_\ell+j} - \hat{x}(k_\ell+j|k_\ell) \\
&= Ax_{k_\ell+j-1} + B\hat{u}(k_\ell+j-1|k_\ell) + g_{k_\ell+j-1} \\
&- (A\hat{x}(k_\ell+j-1|k_\ell) + B\hat{u}(k_\ell+j-1|k_\ell) + \hat{g}_{k_\ell+j-1}) \\
&= Ax_{k_\ell+j-1} - \hat{x}(k_\ell+j-1|k_\ell) + g_{k_\ell+j-1} - \hat{g}_{k_\ell+j-1} \\
&= A\hat{e}_{k_\ell+j-1} + g_{k_\ell+j-1} - \hat{g}_{k_\ell+j-1}.
\end{aligned} \tag{16}$$

From (12), we can confirm that

$$\|\bar{x}(k_\ell+j-1) - \hat{x}(k_\ell+j-1|k_\ell)\| < \|GC\| \cdot e_{\mathrm{trg}}, \tag{17}$$

as long as the event does not occur upto the point of $k_\ell + j - 1$. As follows, the observer error $\hat{e}_{k_\ell+j-1}$ in (16) can be represented by $\bar{x}(k_\ell+j-1) - \hat{x}(k_\ell+j-1|k_\ell)$, given by:

$$\begin{aligned}
&\bar{x}(k_\ell+j-1) - \hat{x}(k_\ell+j-1|k_\ell) \\
&= (A-GC)\hat{x}(k_\ell+j-2|k_\ell) + B\hat{u}(k_\ell+j-2|k_\ell) \\
&+ \hat{g}_{k_\ell+j-2} + GCx_{k_\ell+j-1} \\
&- (A\hat{x}(k_\ell+j-2|k_\ell) + B\hat{u}(k_\ell+j-2|k_\ell) + \hat{g}_{k_\ell+j-2}) \\
&= GC\left(x_{k_\ell+j-1} - \hat{x}(k_\ell+j-1|k_\ell)\right) \\
&= GC\hat{e}_{k_\ell+j-1}.
\end{aligned} \tag{18}$$

Based on (17) and (18), we can have the relationship:

$$\begin{aligned}
&\|\bar{x}(k_\ell+j-1) - \hat{x}(k_\ell+j-1|k_\ell)\| \\
&= \|GC\hat{e}_{k_\ell+j-1}\| \leq \|GC\| \cdot \|\hat{e}_{k_\ell+j-1}\| < \|GC\| \cdot e_{\mathrm{trg}}.
\end{aligned} \tag{19}$$

Therefore, we have

$$\|\hat{e}_{k_\ell+j-1}\| < e_{\mathrm{trg}},$$

and the expectation of the prediction error at the next sampling instatnt can be defined by

$$\begin{aligned}
\mathbb{E}[\|\hat{e}_{k_\ell+j}\|] &\leq \mathbb{E}[\|A\| \cdot e_{\mathrm{trg}} + g_{k_\ell+j-1} - \hat{g}_{k_\ell+j-1}] \\
&\leq \|A\| \cdot e_{\mathrm{trg}} + e_g.
\end{aligned} \tag{20}$$

■

Assumption 1: The MPC optimization in (8) based on the event-triggered policy in (12) is feasible for all the constraints. Also, there exists the feedback gain K according to [1], such that $\lim_{k\to\infty} \|M(\hat{x}(k|k_\ell)) - K\hat{x}(k|k_\ell)\| = 0$, where $A + BK$ is Schur and $M(\hat{x}(k|k_\ell))$ is the optimal control input obtained from the MPC, and K is given by:

$$K = -(B^T Q_N B + R)^{-1} B^T Q_N, \tag{21}$$

and

$$Q_N = A^T Q_N A + Q - A^T Q_N B(B^T Q_N B + R)^{-1} B^T Q_N A. \tag{22}$$

Theorem 2: If *Assumption 1* is satisfied, the the closed-loop system is bounded such that

$$\begin{aligned}
\lim_{k\to\infty} \mathbb{E}[\|x_k\|] &\leq \sup_{g\in\mathcal{G}} \|\sum_{j=0}^{\infty} \bar{A}^j g(x,u)\| \\
&+ \lim_{k\to\infty} \sum_{j=0}^{k-1} \|\bar{A}^j BK\| e_{\max},
\end{aligned} \tag{23}$$

where $\bar{A} = (A + BK)$.

Proof: The closed-loop system is given by

$$\begin{aligned}
x_{k_\ell+1} &= Ax_{k_\ell} + B\left(M(\hat{x}(k_\ell|k_\ell)) - K\hat{x}(k_\ell|k_\ell)\right) \\
&+ BK\hat{x}(k_\ell|k_\ell)) + g\left(x_{k_\ell}, M(\hat{x}(k_\ell|k_\ell))\right) \\
&= (A+BK)x_{k_\ell} + B\left(M(\hat{x}(k_\ell|k_\ell)) - K\hat{x}(k_\ell|k_\ell)\right) \\
&- BK\hat{e}_{k_\ell} + g(x_{k_\ell}, M(\hat{x}(k_\ell|k_\ell))).
\end{aligned}$$

Define $(A + BK) = \bar{A}$ and $\bar{k} = k_\ell + k - 1 - j$, then we can have the state evolution in the following:

$$\begin{aligned}
&\lim_{k\to\infty} \mathbb{E}[\|x_k\|] \\
&= \lim_{k\to\infty} \mathbb{E}\Big[\|\bar{A}^k x_{k_\ell} + \sum_{j=0}^{k-1} \bar{A}^j B\left(M(\hat{x}(\bar{k}|k_\ell)) - K\hat{x}(\bar{k}|k_\ell)\right) \\
&- \sum_{j=0}^{k-1} \bar{A}^j BK\hat{e}_{\bar{k}} + \sum_{j=0}^{k-1} \bar{A}^j g(x_{\bar{k}}, M(\hat{x}(\bar{k}|k_\ell)))\|\Big] \\
&= \lim_{k\to\infty} \mathbb{E}\Big[\| - \sum_{j=0}^{k-1} \bar{A}^j BK\hat{e}_{\bar{k}} + \sum_{j=0}^{k-1} \bar{A}^j g(x_{\bar{k}}, M(\hat{x}(\bar{k}|k_\ell)))\|\Big],
\end{aligned} \tag{24}$$

where (24) is obtained by *Assumption 1*. As a result, the state convergence is given by:

$$\lim_{k\to\infty} \mathbb{E}[\|x_k\|] \leq r + \lim_{k\to\infty} \sum_{j=0}^{k-1} \|\bar{A}^j BK\| e_{\max},$$

where $r = \sup_{g\in\mathcal{G}} \|\sum_{j=0}^{\infty} \bar{A}^j g(x,u)\|$ ■

IV. ERM Learning

The role of machine learning in this paper has two-folds whose first is the design of disturbance predictor $\hat{g}$ in (5) and second is the finding error bound e_g as described in (15). Section IV-A establishes g_k and Section IV-B establishes e_g by applying ERM with a kernel regression. The basic notions of statistical learning used in the paper can be found in any textbook on advanced probability, for instance [14].

A. Learning formulation

Define $g_k \equiv g(x_k, u_k)$. The learning error in (15) can be stated by:

$$\begin{aligned}
&\mathbb{E}[|g_k - \hat{g}_k|] \leq \sqrt{\mathbb{E}[|g_k - \hat{g}_k|^2]} \\
&= \sqrt{\mathbb{E}[(g_k(1) - \hat{g}_k(1))^2] + \cdots + \mathbb{E}[(g_k(d) - \hat{g}_k(d))^2]} \\
&\leq \sqrt{b(1) + b(2) + \cdots + b(d)} \\
&= e_g,
\end{aligned} \tag{25}$$

where $g_k(j), \hat{g}_k(j) : \mathbb{R}^{d+m} \to \mathbb{R}$. The bound e_g can be defined as the sum of the upper bounds $b(j) > 0$ for $j = 1, \ldots, d$. ERM learning focuses on finding an upper bound of one of $b(j)$ by omitting the notation, such that

$$\mathbb{E}[(g - \hat{g})^2] \leq b, \tag{26}$$

where $g, \hat{g} : \mathbb{R}^{d+m} \to \mathbb{R}$, and $b > 0$.

Suppose that we are given a set of i.i.d. training data set $\{(X_i, Y_i)\}_{i=1}^n$ with $X \in \chi \subset \mathbb{R}^{d+m}, Y \in \mathcal{Y} \subset \mathbb{R}$, where the pair of random variables (X, Y) follow the unknown distribution $\mathbb{P}_{XY}$. The goal of the learning is to use the training data to find a prediction rule $g : \chi \to \mathcal{Y}$ that reduces the empirical risk, which is given by:

$$\hat{R}_n(g) := \mathbb{E}[l(g(X), Y)|g], \tag{27}$$

where $l : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ is loss function.

Suppose that there is a countable, possibly infinite set of candidate prediction rules $g \in \mathcal{G}$, then the peneralized empirical risk minimization chooses the one that minimizes the empirical risk such that

$$\hat{g} = \underset{g \in \mathcal{G}}{\operatorname{argmin}} \left(\hat{R}_n(g) + C(g, n, \delta) \right), \tag{28}$$

where $C(g, n, \delta)$ is the penalty cost that is a function of the model g, number of samples n, and a parameter $\delta \in (0, 1)$. It will be defined in (36).

Now, we look into the risk error bound of the predictor $\hat{g}_n$. Let $l : \mathcal{Y} \times \mathcal{Y} \to [0, B]$ be a bounded loss function. Then, with propability at least $1 - \delta$, empirical risk is bounded such that

$$\begin{aligned}
R(\hat{g}) - \inf_{g \in \mathcal{G}} R(g) &\leq C(g, n, \delta) \\
&= 2B \sqrt{\frac{\log\left(1 + \exp^{c(g)}\right) - \log(\delta)}{2n}},
\end{aligned} \tag{29}$$

where $\inf_{g \in \mathcal{G}} R(g)$ is the Bayes risk for all possible predictions and $c(g)$ is a function indicates the model complexity of $g(\cdot)$.

Eqn. (29) states the theoretical guarantee about the boundness of the prediction error for the unknown target function, once we find the empirical risk minimizer by using (28). The bound is controlled by the model compelexity function $c(g)$, the number of training data n, the parameter δ.

B. A penalized empirical risk minimization using kernel regression

We consider a kernel regression model, given by

$$g(x) = \mathbb{E}[Y = y | X = x, \omega] = \omega^T K_x, \tag{30}$$

where $\omega \in \mathbb{R}^n$ is to be estimated, and $K_x \in \mathbb{R}^n$ is a kernel vector whose i-th element corresponding to the i-th training point X_i is as follows

$$K_x(i) = \exp^{-(X_i - x)^2 / (2\Sigma^2)}. \tag{31}$$

We assume that the likelihood follows Gaussian distribution:

$$\mathbb{P}(Y = y | X = x, \omega) = \frac{1}{\sqrt{2\pi\sigma_1^2}} \exp^{-(\omega^T K_x - y)^2 / (2\sigma_1^2)}, \tag{32}$$

and the prior also follows Gaussian distribution such that

$$\mathbb{P}(\omega) = \frac{1}{\sqrt{2\pi\sigma_2^2}} \exp^{-(\omega^T \omega) / (2\sigma_2^2)}. \tag{33}$$

Accordingly, the empirical risk function in (27) can be defined:

$$\begin{aligned}
\hat{R}_n(g) &= \frac{1}{n} \sum_{i=1}^n l(g(x_i), Y_i) = \frac{1}{n} \sum_{i=1}^n -\log \mathbb{P}(Y_i | X_i, \omega) \\
&= \frac{1}{2\sigma_1^2 n} \sum_{i=1}^n (\omega^T K_{X_i} - Y_i)^2 + \frac{1}{2} \log 2\pi\sigma_1^2.
\end{aligned} \tag{34}$$

The model complexity function $c(g)$ as the negative log of prior probability of g is given by

$$\begin{aligned}
c(g) &= -\log\left(\mathbb{P}(g)\right) = -\log\left(\mathbb{P}(\omega^T K_X)\right) = -\log\left(\mathbb{P}(\omega)\right) \\
&= \frac{1}{2} \log(2\pi\sigma_2^2) + \frac{1}{2} \frac{\omega^T \omega}{\sigma_2^2}.
\end{aligned} \tag{35}$$

Accordingly, $C(g, n, \delta)$ is given by:

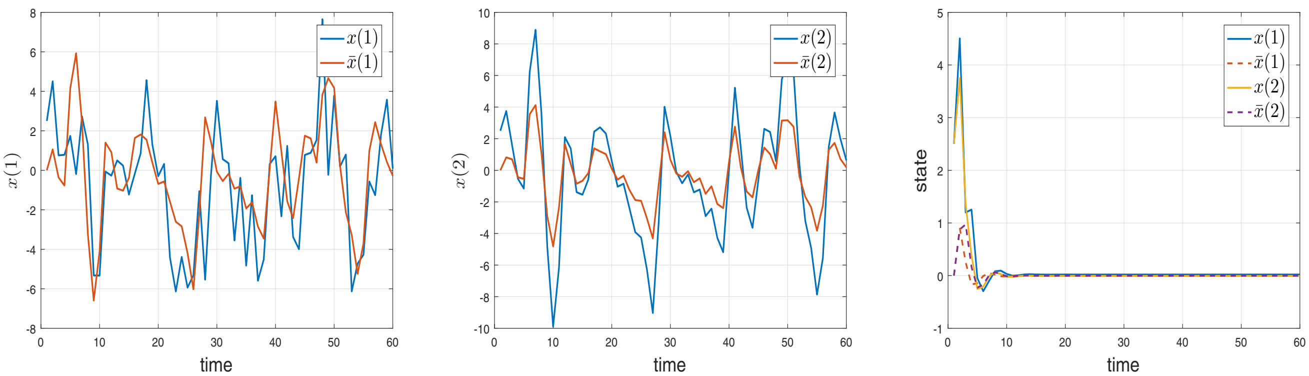
$$\begin{aligned}
&C(g, n, \delta) = \\
&2B \sqrt{\frac{\log(1 + \exp^{0.5} + 2\pi\sigma_2^2 + \exp^{-(\omega^T \omega)/(2\sigma_2^2)}) - \log(\delta)}{2n}}
\end{aligned} \tag{36}$$

As a result, the peneralized empirical risk minimization in (28) is specified with the kernel regression in (30), given by

$$\hat{\omega} = \underset{\omega}{\operatorname{argmin}} \left(\frac{1}{2\sigma_1^2 n} \sum_{i=1}^n (\omega^T K_{X_i} - Y_i)^2 + C(g, n, \delta) \right). \tag{37}$$

Also, by inserting (35) into (29), we can obtain the error bound of the predictor $\hat{g} = \hat{\omega}^T K_X$. Gradient descent method can be used to solve (37).

Finally, we need to find the relationship between the disturbance prediction error bound in (26) and the expected risk bound in (29). From the fact that the regression function



(a) without compensation (b) without compensation (c) with ERM learnin based disturbance attenuation

Fig. 2: Time-triggered MPC with/without ERM learning

in (30) has the smallest possible risk, the expected risk error is equivalent to the expected prediction error such that

$$\mathbb{E}[R(\hat{g})] - \inf_{g\in\mathcal{G}} R(g) = \mathbb{E}[(g-\hat{g})^2], \tag{38}$$

where w is the target to be estimated by the learning model $\hat{g}$. As a result, after obtaining the solution $\hat{w}$ by (37), the bound in (26) becomes

$$\mathbb{E}[(g-\hat{g})]^2 \le b = C(g,n,\delta). \tag{39}$$

V. Simulation

For the simulation study, the following system model is considered:

$$x_{k+1} = \begin{bmatrix} 1 & -0.5 \\ 0.5 & 0 \end{bmatrix} x_k + \begin{bmatrix} 0.5 \\ 0 \end{bmatrix} u_k \tag{40}$$
$$+ \begin{bmatrix} x(2) + 2\sin(u(1)) + 2\sin(x(1)) \\ x(1) \end{bmatrix} + w_k, \tag{41}$$
$$y_k = \begin{bmatrix} 0.5 & 0 \end{bmatrix} x_k,$$

where w_k is defined as additive white Gaussian noise with standard deviation 0.01. The constraints of the state and input are given by $|x(1)|, |x(2)| < 10$ and $|u| < 2$. The running cost functions are given by $Q = 2I_{2\times 2}$ and $R = 0.1$. The terminal cost function Q_N is chosen by *Assumption 1*, given by

$$Q_N = \begin{bmatrix} 1.554 & -0.151 \\ -0.151 & 1.080 \end{bmatrix},$$

with the state-feedback gain $K = [1.513 \quad -0.795]$. The prediction horizon is set to $N = 10$ steps, and the time interval is set to $T = 1$ sec. The initial positions set to $x_0 = [2; 3]$. The event threshold for the event-triggered implementation in (12) is set such that $\|GC\| \cdot e_{\text{trg}} = 0.5$.

In order to learn system, training data is collected by iterative simulation tests and recording all of the measured states and control inputs as well as true states. For creating training data sets, we repeated control operations for different initial points. Let us define k_i as the discrete time step with the index i on training phase, X_i as the training input set, and Y_i as the training output. Then, the training data set $\{(X_i, Y_i)\}_{i=1}^n$ is defined by

$$X_i = [x_{k_i}^T, u_{k_i}^T]^T \in \mathbb{R}^{d+m},$$
$$Y_i = x_{k_i+1} - (Ax_{k_i} + Bu_{k_i} + w_{k_i}) \in \mathbb{R},$$

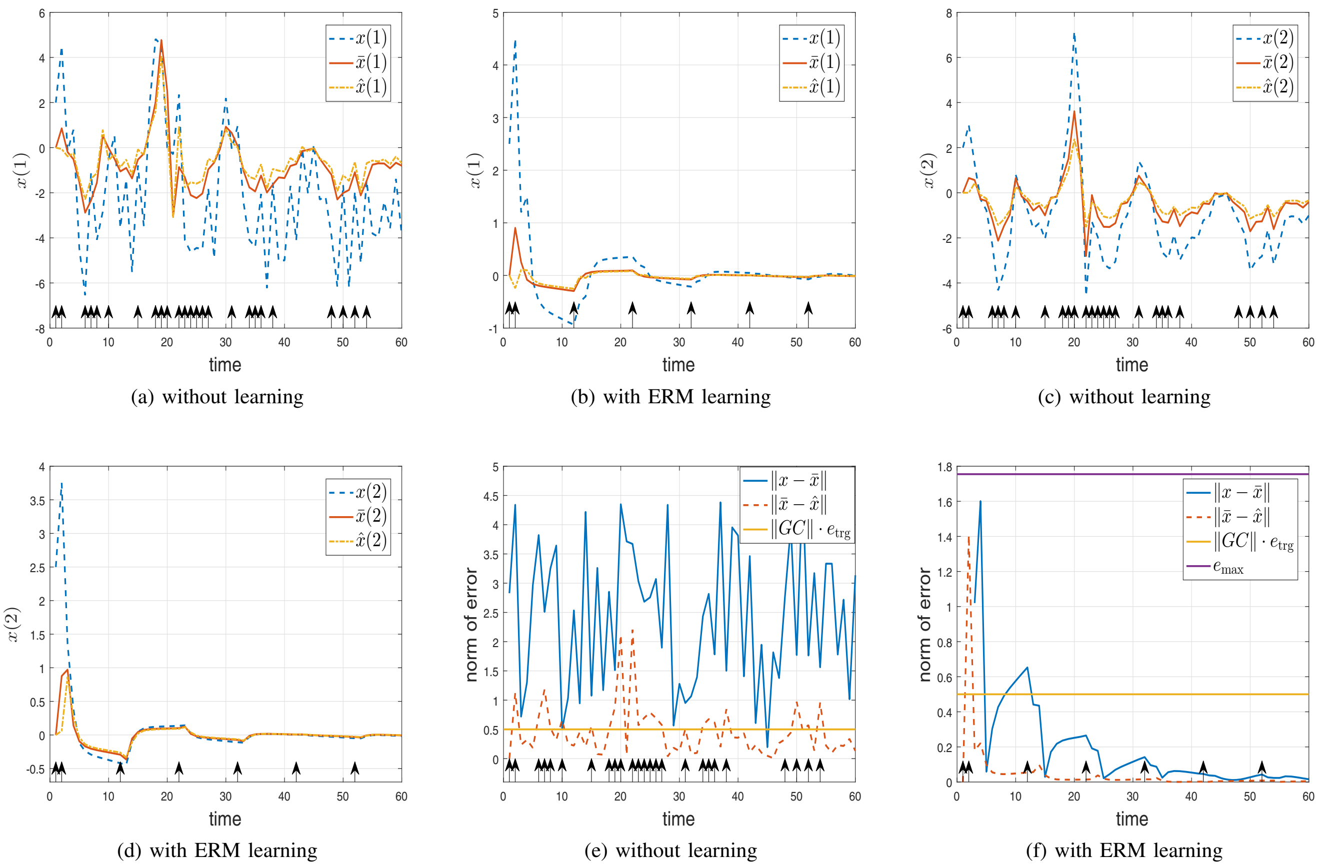
where A and B are defined in (40).

Fig. 2 shows the estimation results where Figs. 2(a) and 2(b) are without the learning and Fig. 2(c) is the result when the ERM learning is applied. The nonlinear disturbance makes oscillatory state variations as in Figs. 2(a) and 2(b). However, accurate estimation by the learning vanishes them as shown in Fig. 2(c). This estimation comparison causes the difference of event-triggered control results in Fig. 3.

The true state and the estimated states are shown in Figs. 3(a), 3(b), 3(c), and 3(d). When the learning-based estimation is applied, the regulation performance is much better. Also, in case of Figs. 3(b) and 3(d), which are the PAC learning is used, it is noted that the triggering instants only occur when the horizon $N = 10$ is expired. On the other hand, without the learning in case of Figs. 3(a) and 3(c), the triggering instances happen more often. This can be confirmed by Figs. 3(e) and 3(f), which show the error of estimates and the threshold of the trigger. The trigger occurs whenever the error $\|\bar{x} - \hat{x}\|$ exceeds in Fig. 3(e), while a few triggers are caused by the lapse of the horizon in Fig. 3(f). Also, the error bound value $e_{\max}$ is kept above the true error $\|x - \bar{x}\|$.

VI. Conclusion

This paper presented the event-based MPC approach based on ERM learning-based estimator. Disturbance is compensated by the learning capability. The ERM learning error bound enables to analyze the stability of the control system with respect to the learning properties. Simulation results showed that the developed control scheme yields effective event-triggering policy while guaranteeing a desired performance.



(a) without learning (b) with ERM learning (c) without learning

(d) with ERM learning (e) without learning (f) with ERM learning

Fig. 3: Event-triggered MPC with/without ERM learning

REFERENCES

[1] D. Lehmann, E. Henriksson, and K. H. Johansson, "Event-triggered model predictive control of discrete-time linear systems subject to disturbances," in *European Control Conference*, 2013, pp. 1156–1161.

[2] J. Yoo, A. Molin, M. Jafarian, H. Esen, D. V. Dimarogonas, and K. H. Johansson, "Event-triggered model predictive control with machine learning for compensation of model uncertainties," in *IEEE Conference on Decision and Control*, 2017, to be publisehd.

[3] J. Sjöberg, Q. Zhang, L. Ljung, A. Benveniste, B. Delyon, P.-Y. Glorennec, H. Hjalmarsson, and A. Juditsky, "Nonlinear black-box modeling in system identification: a unified overview," *Automatica*, vol. 31, no. 12, pp. 1691–1724, 1995.

[4] C. P. Chen, Y.-J. Liu, and G.-X. Wen, "Fuzzy neural network-based adaptive control for a class of uncertain nonlinear stochastic systems," *IEEE Transactions on Cybernetics*, vol. 44, no. 5, pp. 583–593, 2014.

[5] G.-X. Wen, C. P. Chen, Y.-J. Liu, and Z. Liu, "Neural-network-based adaptive leader-following consensus control for second-order non-linear multi-agent systems," *IET Control Theory & Applications*, vol. 9, no. 13, pp. 1927–1934, 2015.

[6] A. K. Akametalu, J. F. Fisac, J. H. Gillula, S. Kaynama, M. N. Zeilinger, and C. J. Tomlin, "Reachability-based safe learning with gaussian processes," in *IEEE Conference on Decision and Control*, 2014, pp. 1424–1431.

[7] F. Berkenkamp, A. P. Schoellig, and A. Krause, "Safe controller optimization for quadrotors with gaussian processes," in *IEEE International Conference on Robotics and Automation*, 2016, pp. 491–496.

[8] B. Doroodgar, Y. Liu, and G. Nejat, "A learning-based semi-autonomous controller for robotic exploration of unknown disaster scenes while searching for victims," *IEEE Transactions on Cybernetics*, vol. 44, no. 12, pp. 2719–2732, 2014.

[9] A. Punjani and P. Abbeel, "Deep learning helicopter dynamics models," in *IEEE International Conference on Robotics and Automation*, 2015, pp. 3223–3230.

[10] S. Bansal, A. K. Akametalu, F. J. Jiang, F. Laine, and C. J. Tomlin, "Learning quadrotor dynamics using neural network for flight control," in *IEEE Conference on Decision and Control*, 2016, pp. 4653–4660.

[11] V. Vapnik, *The nature of statistical learning theory*. Springer science & business media, 2013.

[12] D. Lehmann and J. Lunze, "Event-based output-feedback control," in *Mediterranean Conference on Control & Automation*, 2011, pp. 982–987.

[13] S. Durand, L. Torres, and J. F. Guerrero-Castellanos, "Event-triggered observer-based output-feedback stabilization of linear system with communication delays in the measurements," in *European Control Conference*, 2014, pp. 666–671.

[14] V. N. Vapnik, "An overview of statistical learning theory," *IEEE transactions on neural networks*, vol. 10, no. 5, pp. 988–999, 1999.