Event-triggered Model Predictive Control with Machine Learning for Compensation of Model Uncertainties

Jaehyun Yoo, Adam Molin, Matin Jafarian, Hasan Esen, Dimos V. Dimarogonas, and Karl H. Johansson

Abstract— **As one of the extensions of model predictive control (MPC), event-triggered MPC takes advantage of the reduction of control updates. However, approaches to event-triggered MPCs may be subject to frequent event-triggering instants in the presence of large disturbances. Motivated by this, this paper suggests an application of machine learning to this control method in order to learn a compensation model for disturbance attenuation. The suggested method improves both event-triggering policy efficiency and control accuracy compared to previous approaches to event-triggered MPCs. We employ the radial basis function (RBF) kernel based machine learning technique. By the universal approximation property of the RBF, which imposes an upper bound on the training error, we can present the stability analysis of the learning-aided control system. The proposed algorithm is evaluated by means of position control of a nonholonomic robot subject to state-dependent disturbances. Simulation results show that the developed method yields not only two times less event triggering instants, but also improved tracking performance.**

I. Introduction

Model predictive control (MPC), which is one of modern control methods, derives control action by solving an optimal control problem based on the information about the dynamics and the current state of the plant. On the other hand, event-triggered control is a sampled data control scheme and requires the executions only when the desired control specification cannot be guaranteed. The application of event-triggered strategies to model predictive control (MPC) has been drawing increasing attention, due to advantages of the reduction of control updates and the capability to deal with nonlinearities and constraints.

The assumption used for general event-triggered policies is the input-to-state stability (ISS) of the plant. The ISS property is used to find sufficient conditions for triggering, in the case of uncertain nonlinear systems with additive disturbance [1]–[4], where an upper bound of the disturbance is known. Maximum upper bound on the disturbance is the major criterion for judging an event-trigger. The major adverse aspect of these event-triggered MPCs is that the triggering occurs too frequently when the bound is large, which is a common situation in practice.

*This work has been supported in part by the Knut and Alice Wallenberg Foundation, the Swedish Research Council (VR), the Swedish Foundation for Strategic Research including the SSF-NRF Sweden-Korea research program, and the Denso AUTOMOTIVE GmbH.

J. Yoo, M. Jafarian, D. V. Dimarogonas, and K. H. Johansson are with the ACCESS Linnaeus Center and the School of Electrical Engineering, KTH Royal Institute of Technology, Stockholm, Sweden, Emails: {jaehyun,matinj,dimos,kallej@kth.se}

A. Molin and H. Esen are with the Technical Research Department, DENSO AUTOMOTIVE Deutschland GmbH, Freisinger Str. 21, 85386 Eching, Germany, Emails: {a.molin,h.esen@denso-auto.de}

Motivated by this problem, the main contribution of this paper is applying machine learning technique to compensate for the disturbance to improve triggering efficiency as well as control performance. Many of disturbance rejection methods such as adaptive control require restrictions on the disturbance such as a known upper bound or a known structure (e.g, constant or harmonic) [5]–[7]. The suggested learning-aided control approach can relax these restrictions and automatically model a compensator.

Many works have applied machine learning techniques to enhance performance of control systems by learning empirical model synthesis. In [8], [9], the unknown part of a dynamic model or disturbance is modelled by supervised learning. In [10]–[12], reinforcement learning is used to derive a control policy for a system with unknown dynamics through trial-and-error interactions in a dynamic environment. In [13], [14], parametric controllers are automatically designed by machine learning technique while guaranteeing safety of robotic systems. Despite their efforts to exploit machine learning methods, they have only focused on improving tracking performance, but not provided stability analysis.

In this paper, the distinctive feature is that we are capable of conducting a stability analysis despite the fact that the control design is based on machine learning techniques. To this purpose, we adopt the radial basis function (RBF) as a kernel function that is a part of the the machine learning algorithm. As an RBF characteristics, it has the universial approximation property that serves as an upper bound on the training error. This property is used to guarantee stability of the suggested control system.

As one of RBF-based machine learning algorithms, we employ least square support vector regression (LSSVR). Support vector machine (SVM) [15] has been widely used in a range of applications such as data mining, classification, regression and time-series forecasting [16]. The LSSVR developed from [17] is modified from the standard SVM. This reformulation simplifies the optimization in a way that the solution is characterized by a linear system while standard SVM solves an optimization in every iteration step by interior point methods. In particular, when machine learning is used for a control system, computational issues are important because control inputs should be properly provided in time. Thus, LSSVR can be one of suitable applications to combine with control systems.

The proposed algorithm is evaluated to position control of a nonholonomic robot and is compared to a standard event-triggered MPC. The simulation results show the learning effect on control performance, triggering condition, and disturbance compensation under the event-triggered MPC framework. Also, we can see that smooth control inputs are generated from the suggested control method, while oscillated control inputs are shown for a standard event-triggered MPC scheme. As a result, the developed method yields not only two times less event triggering instants, but also improved tracking performance.

The rest of this paper is organized as follows. Section II presents the system description and summarizes contributions. Section III describes the control problem formulation under nonlinear MPC with some assumptions. Section IV introduces the suggested learning-based event-triggered MPC. Section V describes the LSSVR machine learning algorithm. Section VI and Section VII show simulation results and concluding remarks.

II. System description

Consider the nonlinear discrete-time dynamic system

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

where $x_k \in \mathbb{R}^n$ denotes the system state, $u_k \in \mathbb{R}^m$ is the control input, and $d_k \in \mathbb{R}^n$ is the additive disturbance. The system is subject to constraints on the state and on the control variables $x_k \in \mathcal{X},\ u_k \in \mathcal{U}$.

We assume that the uncertainty d_k is a function of x_k and u_k satisfying:

$$x_{k+1} = f(x_k, u_k) + d(x_k, u_k), \tag{2}$$

with known $f(x_k, u_k)$ and unknown disturbance $d(x_k, u_k)$.

This paper uses model predictive control (MPC) for tracking. The control input sequence $u(k+j|k)$ is calculated over a finite horizon $0 \leq j \leq N$.

To counteract uncertainty $d(x_k, u_k)$, we aim to design the disturbance compensator $g(x_k, u_k)$ such that

$$\|d(x_k, u_k) - g(x_k, u_k)\| = \varepsilon_k \leq \varepsilon, \tag{3}$$

where the uncertainty compensation error is bounded by $\varepsilon > 0$, and we assume that the full state x_k is measurable. The main contribution of this paper is applying a machine learning technique to model $g(x_k, u_k)$. The bound in (3) can be obtained by the property of the universal approximation [18] when applying the radial basis function (RBF) as a kernel function used in the machine learning algorithm.

The predictive state sequence should be estimated for calculating the optimal control in the MPC scheme. Taking the model $g(x_k, u_k)$ into account, the predicted state $\hat{x}(k+j|k)$ for $j \geq 1$ is given by

$$\begin{aligned} \hat{x}(k+j|k) &= f(\hat{x}(k+j-1|k), u(k+j-1|k)) \\ &+ g(\hat{x}(k+j-1|k), u(k+j-1|k)). \end{aligned} \tag{4}$$

This prediction model in (4) is different to the prediction model of the previous works [1]–[3] which use only the nominal model, i.e the first term of the right hand in (4).

The extension to event-triggered MPC is considered with disturbance compensator. The idea behind the event-triggered MPC is to trigger a solution process of the optimal control problem, only when it is needed. Otherwise, it keeps using the control input calculated at the last event-trigger instant. The updates of the control law depend on the error of the actual and the predicted states of the system. Therefore, the estimation by the disturbance compensation in (4) affects the triggering condition as well as the control input calculation.

III. Control problem formulation

In this section, the problem formulation of nonlinear MPC is presented. The design and analysis of the proposed controller is provided along with some assumptions that are necessary to achieve stability of the closed-loop system.

A. System assumptions

Assumption 1: It is assumed that $f(0,0) = 0$ and $f(x,u)$ is locally Lipschitz in the domain $\mathcal{X} \times \mathcal{U}$ with Lipschitz constant L_f.

Let us define the preidction model $\hat{f}(x_k, u_k) = f(x_k, u_k) + g(x_k, u_k)$ from (4). Then $\hat{f}(x_k, u_k)$ is also locally Lipschitz, given by:

$$\|\hat{f}(x_1, u) - \hat{f}(x_2, u)\| \leq L_{\hat{f}} \|x_1 - x_2\|. \tag{5}$$

Remark 1: Lipschitz assumption for the nominal model f with Lipschitz constraint L_f is standard for guaranteeing ISS (input-to-state stability). Without loss of generality, the RBF function g satisfies Lipschitz condition with constraint L_g. Because both f and g satisfy Lipschitz condition, a linear combination of them, i.e., $\hat{f}$, is also Lipschitz with Lipschitz constraint $L_{\hat{f}} = L_f + L_g$.

Remark 2: The event-triggered MPC algorithms presented in [1]–[3] assume that the addictive disturbance is bounded such that

$$\|d_k\| \leq \gamma. \tag{6}$$

After applying a designed compensator, the bound of the disturbnace error ε can be smaller than γ:

$$\varepsilon \ll \gamma. \tag{7}$$

The more efficient event-triggering condition is derived by the relationship (7) in comparison with the existing works that should set the bound γ in (6) high enough in practice. In comparison with the existing works, the machine learning based adaptive method can compensate for the uncertainty regardless of its magnitude and structure once its pattern is recognized.

We note e as the Euclidean norm of difference between the true state and the predicted state, given by

$$e(k+j|k) = \|x_{k+j} - \hat{x}(k+j|k)\|. \tag{8}$$

Note that at the current time step k, $e(k|k-1)$ is measurable, while $e(k+j|k-1)$ for $j \geq 1$ are non-measurable predictions. Both measurable and non-measurable variables are necessary to derive the event-triggering condition. The non-measurable prediction has a bound as in the following lemma.

Lemma 1: The error $e(k+j|k)$ for $j \geq 1$ is bounded by

$$
\begin{aligned}
e(k+j|k) &\leq L_f^{j-1}\cdot e(k+1|k)+\frac{L_f^{j-1}-1}{L_f-1}\cdot\varepsilon \\
&\leq \frac{L_f^{j}-1}{L_f-1}\cdot\varepsilon
\end{aligned} \quad (9)
$$

Proof: By using the triangle inequality and the Lipschitz condition of the model, we can obtain the following recursion

$$
\begin{aligned}
\cdot \quad & \|x_{k+1}-\hat{x}(k+1|k)\| \\
& = \|x_{k+1}-f(\hat{x}(k|k),u_k)-g(\hat{x}(k|k),u_k)\| \\
& = \|x_{k+1}-f(x_k,u_k)-g(x_k,u_k)\| \\
& = \|d(x_k,u_k)-g(x_k,u_k)\| = \varepsilon_k \leq \varepsilon \\
\cdot \quad & \|x_{k+j}-\hat{x}(k+j|k)\| \\
& \leq L_f^{j-1}\cdot\varepsilon_k + L_f^{j-2}\cdot\varepsilon_{k+1}+\cdots L_f^{0}\cdot\varepsilon_{k+j-1} \\
& \leq L_f^{j-1}\cdot e(k+1|k)+\frac{L_f^{j-1}-1}{L_f-1}\cdot\varepsilon \\
& \leq \frac{L_f^{j}-1}{L_f-1}\cdot\varepsilon.
\end{aligned}
$$

■

Lemma 2: The error of predictions between current and previous time steps is bounded such that

$$\|\hat{x}(k+j|k)-\hat{x}(k+j|k-1)\| \leq L_{\hat{f}}^{j} e(k|k-1) \quad (10)$$

Proof:

$$
\begin{aligned}
\cdot \quad & \|\hat{x}(k|k)-\hat{x}(k|k-1)\| = e(k|k-1) \\
\cdot \quad & \|\hat{x}(k+1|k)-\hat{x}(k+1|k-1)\| \\
& = \|f\left(\hat{x}(k|k),u_k\right)+g\left(\hat{x}(k|k),u_k\right) \\
& \quad -f\left(\hat{x}(k|k-1),u_k\right)-g\left(\hat{x}(k|k-1),u_k\right)\| \\
& \leq L_{\hat{f}}\|\hat{x}(k|k)-\hat{x}(k|k-1) = L_{\hat{f}} e(k|k-1) \\
\cdot \quad & \|\hat{x}(k+j|k)-\hat{x}(k+j|k-1)\| \leq L_{\hat{f}}^{j} e(k|k-1).
\end{aligned}
$$

■

Lemma 1 and *Lemma 2* are used to analyze stability and feasibility of the control system. We note that their proofs are similar to the proofs shown in [2]–[4]. Thus, the remaining proofs in this paper will be condensed because of the similarity to the existing works [2]–[4]. However, we highlight the differences to the existing works by *Remarks 2, 3*, and *4*.

B. MPC formulation

MPC is a control strategy that calculates predictions of current and future control inputs by solving an online finite horizon optimal control problem (FHOCP). The current and predictive states and control inputs are denoted in vector format as

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

where N is length of prediction horizon, and the initial state is given such that $\hat{x}(k|k)=x_k$. The FHOCP is formulated in the following:

$$
\begin{aligned}
&\min_{U(k)} J(X(k),U(k)) = \\
&\min_{U(k)} \sum_{i=0}^{N-1} L\left(\hat{x}(k+i|k),u(k+1|k)\right)+V(\hat{x}(k+N|k)),
\end{aligned}
$$

subject to

$$
\begin{aligned}
&\hat{x}(k+j+1|k)=\hat{f}\left(\hat{x}(k+j|k),u(k+j|k)\right)\in\mathcal{X}_j \\
&u(k+j|k)\in\mathcal{U}, \\
&\hat{x}(k+N|k)\in\mathcal{X}_f, \\
&\forall j=0,\ldots,N-1,
\end{aligned} \quad (11)
$$

where $\mathcal{X}_f$ denotes the terminal constraint set, $L(\cdot)$ is the running cost function, and $V(\cdot)$ is the terminal cost function. The following assumptions are required to guarantee control stability and feasibility.

Assumption 2: The running cost $L(x,u)$ is Lipschitz continuous in $\mathcal{X}\times\mathcal{U}$, with a Lipschitz constant L_c. Also, It satisfies $L(0,0)=0$ and there are positive integers $\alpha>0$ and $\omega\geq 1$, such that $L(x,u)\geq\alpha\|(x,u)\|^{\omega}$.

Assumption 3: There exists a local stabilising controller $h(x)$ for the terminal set $\mathcal{X}_f$ in the sense that $V(f(x,h(x)))-V(x)\leq -L(x,h(x))$ for $\forall x\in\Phi$, and V is Lipschitz in Φ with Lipschitz constant L_V. The compact set Φ is given by $\Phi=\{x\in\mathbb{R}^n : V(x)\leq\alpha_\Phi\}$ such that $\Phi\subseteq X_{N-m}$ for $m=1,\ldots,N-1$. The final set $\mathcal{X}_f=\{x\in\mathbb{R}^n : V(x)\leq\alpha_v\}$ is such that $\forall x\in\Phi$, $f(x,h(x))\in\mathcal{X}_f$.

The constraint set $\mathcal{X}_j$ in (11) is made to guarantee that there is a robust positively invariant set for the closed-loop system where a solution of the FHOCP exists. By *Lemma 1*, the set $\mathcal{X}_j$ is defined by

$$
\begin{aligned}
&\mathcal{X}_j=\mathcal{X}\sim\mathcal{B}_j(\varepsilon), \\
&\text{with } \mathcal{B}_j(\varepsilon)=\left\{x\in\mathbb{R}^n : \|x\|\leq\frac{L_f^{j}-1}{L_f-1}\cdot\varepsilon\right\}
\end{aligned} \quad (12)
$$

and the set operator $\sim$ denotes the Pontryagin difference.

Remark 3: From (12), as the set $\mathcal{B}_j(\varepsilon)$ becomes larger, the constraint set $\mathcal{X}_j$ becomes smaller so that feasibility of the FHOCP decreases. Let us call the relationship in (7), which denotes that the set $\mathcal{B}_j(\varepsilon)$ in (12) established by ε is much smaller than the set $\mathcal{B}_j(\gamma)=\left\{x\in\mathbb{R}^n : \|x\|\leq\gamma\cdot(L_f^{j}-1)/(L_f-1)\right\}$ that is defined in the existing works [2]–[4]. Therefore, a degree of the feasibility of the proposed event-triggered MPC is improved as large as the difference between γ and ε.

IV. LSSVR-based Event-triggered MPC

In this section, the triggering condition of the suggested event-triggered MPC control is presented with guarantee of feasibility and stability.

A. Control law and feasibility analysis

In the classic time-triggered MPC strategy, the FHOCP is solved at every time step. In the event-triggered setup, the rest of the predictions might be used until a new event occurs or the lapse of the horizon time.

Suppose that we obtained the optimal solution $U^*(k-1) = \{u^*(k-1|k-1), \dots, u^*(k+N-2|k-1)\}$ and the corresponding optimal cost $J^*(k-1)$ from the last time step $k-1$. Then, the control sequence $U(k+m) = \{u^*(k+m|k+m), \dots, u^*(k+m+N-1|k+m)\}$ for time steps $m = 0, \dots, N-1$ is given by:

$$u^*(k+j|k+m) = \tag{13}$$
$$\begin{cases} u^*(k+j|k-1) & \text{for } j = m, \dots, N-2, \\ h(\hat{x}(k+N-1|k+m)) & \text{for } j = m+N-1. \end{cases}$$

From the control law (13), instead of solving the FHOCP at $k+m$, the control input $\mathbf{U}(k+m)$ is applied to check stability and decide if an event is triggered.

Analysis of feasibility to ensure existence of a solution satisfying all the constraints for the FHOCP, is given the following lemma.

Lemma 3: A system described by (2) is assumed to satisfy all *Assumptions 1-3*. Then, the FHOCP is feasible if the learning error ε is bounded by:

$$\varepsilon \le \frac{(\alpha_\Phi - \alpha_v)}{L_V \cdot L_{\hat{f}}^{N-1}}. \tag{14}$$

Proof: We can follow the feasibilty proof described in [3] by applying *Lemma 1* and *Lemma 2*. ■

B. Stability analysis and triggering condition

Let the optimal cost at time step $k-1$ be $J^*(k-1)$ and the costs of the feasible sequence be indicated by $\bar{J}(k+j)$ for $j = 0, \dots, N-1$. Then, the differences of these costs are given by

$$\Delta J_j = \bar{J}(k+j) - J^*(k-1).$$

Theorem 1: Consider a system described by (2) and assume *Assumptions 1-3*. Then, with the control law (13), ΔJ_j is bounded by:

· for $j = 0$,

$$\Delta J_0 \le L_{Z_0} \cdot e(k|k-1) - \alpha \|x_{k-1}\|^w \tag{15}$$

· for $1 \le j \le N-1$,

$$\Delta J_j \le L_{Z_j} \cdot \frac{L_f^j - 1}{L_f - 1} \cdot \varepsilon - \alpha \sum_{i=0}^{j} \|x_{k+j-i}\|^w, \tag{16}$$

where

$$L_{Z_j} = L_V L_{\hat{f}}^{(N-1)-j} + L_c \frac{L_{\hat{f}}^{(N-1)-j} - 1}{L_{\hat{f}} - 1}. \tag{17}$$

Thus, according to input-to-state stability (ISS) [19], the optimal cost is an ISS-Lyapunov function of the closed loop system. Thus, the closed-loop system is input-to-state stable.

Proof: We can follow the procedures of *Theorems 1* and *2* described in [2] by applying our *Lemma 1* and *Lemma 2*. ■

Consequently, the triggering rule is defined to maintain the stability, which is ensured when ΔJ_i is strictly decreasing:

$$\Delta J_{j+1} \le \Delta J_j. \tag{18}$$

By (15)–(18), the triggering condition can be stated in the following.

Triggering rule:

· for $j = 0$,

$$L_{Z_0} \cdot e(k|k-1) \le \sigma \cdot \alpha \|x_{k-1}\|^w \tag{19}$$

· for $1 \le j \le N-1$,

$$L_{Z_j} \cdot \frac{L_f^j - 1}{L_f - 1} \cdot \varepsilon \le \sigma \cdot \alpha \sum_{i=0}^{j} \|x_{k-i+j}\|^w \tag{20}$$

and

$$\left(L_{Z_j} \cdot \frac{L_f^j - 1}{L_f - 1} - L_{Z_{j-1}} \cdot \frac{L_f^{j-1} - 1}{L_f - 1} \right) \cdot \varepsilon \le \sigma \cdot \alpha \|x_k\|^w, \tag{21}$$

with $0 < \sigma < 1$, and L_{Z_j} from (17).

In case of $j = 0$, the measurable error $e(k|k-1) = \|x_k - \hat{x}(k|k-1)\|$ is used for the triggering condition, where $\hat{x}(k|k-1)$ is obtained by the learning-based estimation. Afterwards, for predictive period $1 \le j \le N-1$, the triggering condition depends on the upper bound ε.

Remark 4: Relatively large values on the left-hand terms in (20) and (21) than the right-hand terms cause frequent triggering instants, which is one of the practical problems in conventional event-triggered MPC [2]–[4]. In this paper, however, the proposition that ε is small enough by uncertainty compensation can prevent the frequent triggering instants.

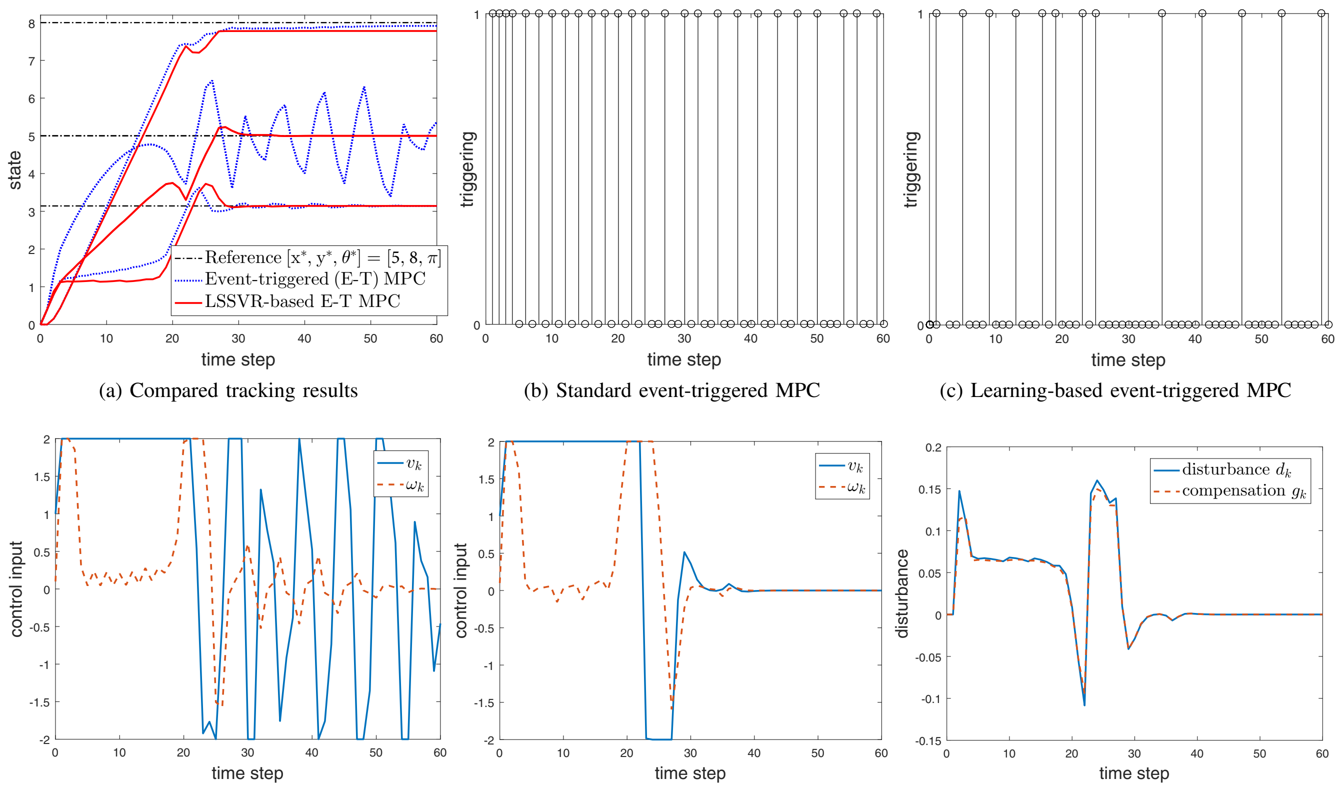
To summarize, suppose that the FHOCP is solved at k. The FHOCP is triggered at $k+j$ if (19) or (20)-(21) is violated. When they are satisfied for all $0 \le j \le N-1$, the FHOCP is triggered at the last horizon step $k+N$.

V. Compensator design based on machine learning

This section describes how to design the compensator $g(x_k, u_k)$ through a machine learning technique.

Machine learning techniques commonly use a kernel function that transforms data in raw representation into feature vector representation, which causes big impact to learning performance. The most popular kernel function is the radial basis function (RBF). In this paper, we employ least square support vector regression (LSSVR) as the learning bodyframe and RBF as the kernel function.

The basic idea of LSSVR is to map the data $\mathbf{x} \in \mathbb{R}^M$ to a higher dimensional feature space $\mathcal{H}$ (reproducing kernel Hilbert space) using a nonlinear mapping $\phi(\mathbf{x}) : \mathbb{R}^M \to \mathcal{H} \in \mathbb{R}^h$, and then find the relationship between the scalar target variable y and the explanatory variable $\mathbf{x}$ (i.e., linear regression) in the kernel space. In other words, given a training set of l training samples $\{(\mathbf{x}_i, y_i)\}_{i=1}^l$, it maps the training samples to a new data set $\{(\phi(\mathbf{x}_i), y_i)\}_{i=1}^l$ with the nonlinear mapping $\phi(\cdot)$.



(a) Compared tracking results (b) Standard event-triggered MPC (c) Learning-based event-triggered MPC

(d) Standard event-triggered MPC (e) Learning-based event-triggered MPC (f) Uncertainty and its estimation

Fig. 1: Simulation results of the proposed algorithm in comparison to the standard event-triggered MPC [2], with respect to (a): tracking results, (b) and (c): event-triggering instants, (d) and (e): control inputs, and (f): uncertainty prediction.

Consider the linear regression model:

$$g(\mathbf{x}) = \langle w, \phi(\mathbf{x}) \rangle + b, \quad w \in \mathbb{R}^h, \; b \in \mathbb{R},$$

where w and b are the coefficients, which are estimated by the following optimization problem:

$$\begin{aligned} \min_{w,b,\xi} \; & \frac{1}{2}\|w\|^2 + c\frac{1}{2}\sum_{i=1}^{l} \xi_i^2, \\ \text{s.t.} \;\; & \xi_i = y_i - g(\mathbf{x}_i), \end{aligned}$$

where the constant $c > 0$ is weight parameter.

A Lagrangian to solve the optimization is given by:

$$\begin{aligned} L(w, b, \xi; \alpha) \;\; &= \;\; \frac{1}{2} w^{\mathrm{T}} w + \frac{1}{2} c \sum_{i=1}^{l} \xi_i^2 \\ & \;\; - \;\; \sum_{i=1}^{l} \alpha_i \left\{ y - \left(w^{\mathrm{T}} \phi(\mathbf{x}_i) + b \right) - \xi_i \right\}, \end{aligned}$$

where $\alpha_i \in \mathbb{R}$ are the Lagrange multipliers, and the KKT (Karush-Kuhn-Tucker) conditions are given by:

$$\begin{aligned} \frac{\partial L}{\partial w} &= 0 \implies w = \sum_{i=1}^{l} \alpha_i \phi(\mathbf{x_i}), \\ \frac{\partial L}{\partial b} &= 0 \implies \sum_{i=1}^{l} \alpha_i = 0, \\ \frac{\partial L}{\partial \xi_i} &= 0 \implies \alpha_i = c\xi_i, \\ \frac{\partial L}{\partial \alpha_i} &= 0 \implies y - \left(w^{\mathrm{T}} \phi(\mathbf{x}_i) + b \right) - \xi_i. \end{aligned}$$

After elimination of the variables w and e, we can obtain the following solution form:

$$\begin{bmatrix} 0 & 1 & \cdots & 1 \\ 1 & k(\mathbf{x}_1, \mathbf{x}_1) + \frac{1}{c} & \cdots & k(\mathbf{x}_1, \mathbf{x}_1) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & k(\mathbf{x}_l, \mathbf{x}_1) + \frac{1}{c} & \cdots & k(\mathbf{x}_l, \mathbf{x}_l) + \frac{1}{c} \end{bmatrix} \begin{bmatrix} b \\ \alpha_1 \\ \vdots \\ \alpha_l \end{bmatrix} = \begin{bmatrix} 0 & y_1 & \cdots & y_l \end{bmatrix}^{\mathrm{T}},$$

where $k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^{\mathrm{T}} \phi(\mathbf{x}_j)$ for $i, j = 1, \ldots, l$ is a kernel function. Finally, by Mercer's theorem [15], the fitting function as the output of LSSVR is given by:

$$g(\mathbf{x}) = \sum_{i=1}^{l} \alpha_i k(\mathbf{x}, \mathbf{x}_i) + b.$$

In this paper, the kernel function is defined by radial basis function (RBF):

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(c_1 \cdot \|\mathbf{x}_i - \mathbf{x}\|/c_2\right),$$

where c_1 and c_2 are weight parameters to control strength and smoothness of the kernel function. Therefore, the LSSVR output is rewritten by:

$$g(\mathbf{x}) = \sum_{i=1}^{l} \alpha_i \exp\left(c_1 \cdot \|\mathbf{x}_i - \mathbf{x}\|/c_2\right) + b. \tag{22}$$

The RBF-based LSSVR (22) has a property of the universal approximation by the following lemma.

Lemma 4: For any given continuous real function $\bar{g}(\mathbf{x})$ on a compact set $\mathcal{L}$ and arbitary $\varepsilon > 0$, there exists on approximation function $g(\mathbf{x})$ formed by (22) such that

$$\sup_{\mathbf{x}\in\mathcal{L}} |\bar{g}(\mathbf{x}) - g(\mathbf{x})| < \varepsilon. \tag{23}$$

Proof: See [18]. ■

VI. Simulation Results

For the simulation study, we consider the position control of a nonholonomic robot including state-dependent disturbance, given by:

$$\begin{aligned} r^{\mathrm{x}}_{k+1} &= r^{\mathrm{x}}_k + (1+\delta) v_k T \cos\theta_k, \\ r^{\mathrm{y}}_{k+1} &= r^{\mathrm{y}}_k + v_k T \sin\theta_k, \\ \theta_{k+1} &= \theta_k + \omega_k T. \end{aligned} \tag{24}$$

In this case, the uncertainty in (2) is defined as $d(x_k, u_k) = \delta v_k T \cos\theta_k$ that is only restricted on x-axis state.

The vector $u = [v, w]$ is the control input. The state vector $x = [r^{\mathrm{x}}, r^{\mathrm{y}}, \theta]^{\mathrm{T}}$ is comprised by the 2-D position of the robot $(r^{\mathrm{x}}, r^{\mathrm{y}})$ and the orientation θ. The uncertainty $\delta = 0.4$ is assumed constant. The constraints of the state and input are given by $|r^{\mathrm{x}}|, |r^{\mathrm{y}}| < 10$ and $|v|, |w| < 2$. Given reference r^*, the running cost function and terminal cost function are given by $L = \|x - r^*\|^2_{Q_1} + \|u\|^2_{Q_2}$, and $V = \|x\|^2_{Q_3}$ with $Q_1 = 2I_{3\times3}$, $Q_2 = 0.5I_{2\times2}$, and $Q_3 = I_{3\times3}$. The prediction horizon is set to $N = 20$ steps, and the time interval is set to $T = 0.2$ sec. The initial positions is $(r^{\mathrm{x}}_0, r^{\mathrm{y}}_0, \theta_0) = (0, 0, 0)$ and the reference goal is $r^* = (5, 8, \pi)$.

A. Performance analysis

We present some comparison results to the conventional event-triggered MPC [2]. From Fig. 1(a), the proposed control scheme outperforms the compared method due to the learning capability in compensating for the oscillating uncertainty. Figs. 1(b) and 1(c) depict the triggering instants of both algorithms. Among 60 time steps, 26 and 13 triggering instants occur, respectively. From Figs. 1(d) and 1(e), the control inputs are shown. The suggested method yields the smooth control action, while the compared method barely approaches the reference by the oscillating control input. The better control performance as well as less triggering instants are caused by the accurate uncertainty estimation as shown in Fig. 1(f).

VII. Conclusion

This paper presented the LSSVR-based event-triggered MPC strategy. It has guaranteed stability and feasibility of the control system. Simulation results showed that the developed control scheme yields two times less event triggering instants and better tracking performance for a unicycle vehicle with model uncertainty, compared with a standard event-triggered MPC.

References

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

[2] A. Eqtami, D. V. Dimarogonas, and K. J. Kyriakopoulos. 'Event-triggered strategies for decentralized model predictive controllers." *IFAC Proceedings, 44.1 (2011): 10068-10073.*

[3] K. Hashimoto, S. Adachi, and D. V. Dimarogonas. "Distributed aperiodic model predictive control for multi-agent systems." *IET Control Theory and Applications 9.1 (2014): 10-20.*

[4] D. L. Marruedo, T. Alamo, and E. F. Camacho. "Input-to-state stable MPC for constrained discrete-time nonlinear systems with bounded additive uncertainties." *IEEE Conference on Decision and Control, 2002.*

[5] D. Cabecinhas, R. Cunha, and C. Silvestre. "A nonlinear quadrotor trajectory tracking controller with disturbance rejection." *Control Engineering Practice 26 (2014): 1-10.*

[6] E. Xargay, R. choe, N. Hovakimyan, and I. Kaminer. "Convergence of a PI coordination protocol in networks with switching topology and quantized measurements.", *IEEE Conference on Decision and Control, 2012.*

[7] M. Jafarian. "Robust consensus of unicycles using ternary and hybrid controllers.", *International Journal of Robust and Nonlinear Control (2017).*

[8] A. Punjani and P. Abbeel. "Deep learning helicopter dynamics models." *IEEE International Conference on Robotics and Automation, 2015.*

[9] S. Bansal, A. K. Akametalu, F. J. Jian, F. Laine, and C. J. Tomlin. "Learning quadrotor dynamics using neural network for flight control." *IEEE Conference on Decision and Control, 2016.*

[10] J. Ko, D. J. Klein, D. fox, and D. Haehnel. "Gaussian processes and reinforcement learning for identification and control of an autonomous blimp." *IEEE International Conference on Robotics and Automation, 2007.*

[11] T. Zhang, G. Kahn, S. Levine, and P. Abbeel. "Learning deep control policies for autonomous aerial vehicles with MPC-guided policy search." *IEEE International Conference on Robotics and Automation, 2016.*

[12] D. Sadigh, E. S. Kim, S. Coogan, S. S. Sastry, and S. A. Seshia. "A learning based approach to control synthesis of markov decision processes for linear temporal logic specifications." *IEEE Conference on Decision and Control, 2014.*

[13] F. Berkenkamp, A. P. Schoellig, and A. Krause. "Safe controller optimization for quadrotors with Gaussian processes." *IEEE International Conference on Robotics and Automation, 2016.*

[14] 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." *IEEE Conference on Decision and Control, 2014.*

[15] C. Cortes, and V. Vapnik. "Support-vector networks." *Machine Learning 20.3 (1995): 273-297.*

[16] J. Yoo, W. Kim, and H. J. Kim. "Distributed estimation using online semi-supervised particle filter for mobile sensor networks." *IET Control Theory & Applications 9.3 (2015): 418-427.*

[17] J.A.K. Suykens, T. V. Gestel, and J. D. Brabanter. "Least squares support vector machines." *World Scientific, 2002.*

[18] J. Park and I. W. Sandberg. "Universal approximation using radial-basis-function networks." *Neural Computation 3.2 (1991): 246-257.*

[19] D. Angeli, E. D. Sontag, and Y. Wang. "A characterization of integral input-to-state stability." *IEEE Transactions on Automatic Control 45.6 (2000): 1082-1097.*