Event-triggered Learning

Friedrich Solowjow, Sebastian Trimpe

Intelligent Control Systems Group, Max Planck Institute for Intelligent Systems, Stuttgart, Germany

Abstract

The efficient exchange of information is an essential aspect of intelligent collective behavior. Event-triggered control and estimation achieve some efficiency by replacing continuous data exchange between agents with intermittent, or event-triggered communication. Typically, model-based predictions are used at times of no data transmission, and updates are sent only when the prediction error grows too large. The effectiveness in reducing communication thus strongly depends on the quality of the prediction model. In this article, we propose event-triggered learning as a novel concept to reduce communication even further and to also adapt to changing dynamics. By monitoring the actual communication rate and comparing it to the one that is induced by the model, we detect a mismatch between model and reality and trigger model learning when needed. Specifically, for linear Gaussian dynamics, we derive different classes of learning triggers solely based on a statistical analysis of inter-communication times and formally prove their effectiveness with the aid of concentration inequalities.

Key words: Networked Control Systems; Statistical Analysis, Event-triggered Control.

1 Introduction

Modern communication technology allows for connecting many devices and systems in unprecedented ways. Thus, enabling applications such as mobile sensor networks, distributed robotics, and multi-vehicle systems, often subsumed as networked control systems (NCSs) (Hespanha et al. 2007). Alongside the potential of NCSs come significant challenges for control design such as delayed transmission, packet drops, or limited bandwidth, which originate from the fact that a shared network is used for feedback. Many methods that have been proposed for addressing these challenges rely on accurate dynamics models. For instance, model-based predictions in event-triggered state estimation and control (Heemels et al. 2012, Lemmon 2010, Miskowicz 2015, Shi et al. 2015) can replace periodic communication up to a certain extent. Only if predictions become inaccurate, communication of sensor measurements is indispensable and thus *triggered when necessary.* In this article, we propose to extend the paradigm of event triggering to model learning and introduce the novel idea of *event-triggered learning* (ETL). With ETL, we detect a mismatch between model and true dynamics and trigger identification of a new model whenever needed. We built ETL on top of a typical event-triggered state estimation architecture (see Fig. 1) and show that the new architecture can cope with changing dynamics and yields further communication savings in an NCS.

The ability to learn is thus, a fundamental aspect of future autonomous systems that are facing uncertain and changing environments. However, the process of learning a new model or behavior typically does not come for free but involves a certain cost. For example, gathering informative data can be challenging due to physical constraints, or updating models can require extensive computation. Moreover, learning for autonomous agents often requires exploring new behavior and thus, typically means deviating from nominal or desired behavior. Hence, the question *when to learn* is essential for the efficient operation of autonomous systems. We address this question with ETL, and this article develops the concept specifically in the context of NCSs.

Main Idea: Event-triggered Learning

We explain the main idea of event-triggered learning using the schematic in Fig. 1. The figure depicts a canonical problem, where one agent ('Sending agent' on the left) has information that is relevant for another agent at a different location ('Receiving agent'). For instance, this

⋆ This work was supported in part by the German Research Foundation (DFG) Priority Program 1914 (grant TR 1433/1-1), the International Max Planck Research School for Intelligent Systems (IMPRS-IS), the Cyber Valley Initiative, and the Max Planck Society.

Email addresses: solowjow@is.mpg.de (Friedrich Solowjow), trimpe@is.mpg.de (Sebastian Trimpe).

setting is representative of remote monitoring scenarios, distributed sensor fusion, or two agents of a multi-agent network. For resource-efficient communication, a standard event-triggered state estimation (ETSE) architecture is used (shown in blue). The main contribution of this work is to incorporate learning into the ETSE architecture. By designing an event trigger also for model learning (in green), learning tasks are performed only when necessary. Next, we explain the core components of the proposed framework.

The sending agent in Fig. 1 monitors the state of a dynamic process (either directly measured or obtained via state estimation) and can transmit this state information to the remote agent. The true parameters θ of the process are unknown to both agents. An event-triggered protocol is used to save network resources. The receiving agent uses a model (with parameters $\hat{\theta}$) of the process for predicting the state at times of no communication. The sending agent implements a copy of the same prediction and compares it to the current state in the 'State Trigger', which triggers a state communication whenever the prediction deviates too much from the actual state. This general scheme is standard in ETSE literature (see (Shi et al. 2015, Trimpe 2017, Trimpe & Campi 2015) and references therein). The effectiveness of this scheme will generally depend on the accuracy of the prediction, and, thus, the quality of the model $\hat{\theta}$.

The key idea of this work is to trigger model learning when communication rates deviate significantly from what is expected. Because performing a learning task is costly itself (e.g., involving computation and communication resources, as well as possibly causing deviation from the control objective), we propose event-triggering rules also for model learning ('Learning Trigger'). Newly learned models are then shared with the remote agent to improve its predictions. Since communication itself is triggered by model-based state predictions, we obtain a tractable feature to quantify model accuracy by analyzing the communication pattern. Further, we avoid analyzing raw output data, which is possibly multidimensional and highly correlated. Thus, we propose a method to obtain improved models from data when needed, which leads to superior communication rates.

While the idea of using event triggering to save communication in estimation or control is quite common by now, this work proposes event triggering also on a higher level. Triggering of learning tasks yields improved prediction models, which are the basis for ETSE at the lower level.

Related Work

Various event-triggered control (Heemels et al. 2012, Lemmon 2010, Miskowicz 2015) and state estimation (Lemmon 2010, Shi et al. 2015, Trimpe 2017, Trimpe & Campi 2015) algorithms have been proposed for improving resource usage in NCSs. Approaches differ, among others, in the type of models that are used for predictions. The send-on-delta protocol (Miskowicz 2006) triggers data transmission when the difference between the current and last communicated value passes a threshold. This protocol is extended to linear predictions in (Suh 2007), which are obtained by approximating the signal derivative from data. More elaborate protocols use dynamics models of the observed process, which typically leads to more effective triggering (Battistelli et al. 2018, Han et al. 2015, Sijs et al. 2014, Sijs & Lazar 2012, Trimpe & Baumann 2019, Trimpe & D'Andrea 2011, 2014, Wu et al. 2013).

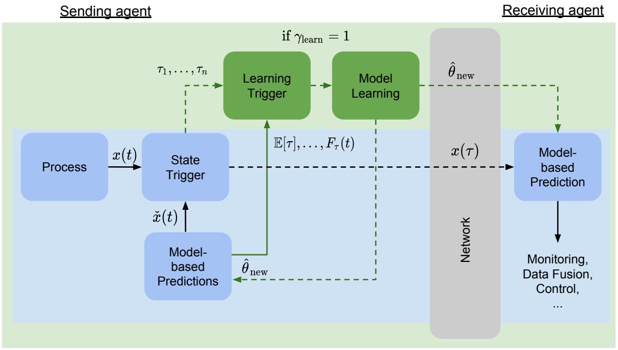


Fig 1. Proposed event-triggered learning architecture for a networked control problem between two agents. Based on a typical event-triggered state estimation architecture (in blue), we propose event-triggered learning (in green) to improve predictions and lower communication between Sending and Receiving agents. Learning experiments are themselves triggered as necessary by comparing empirical with expected inter-communication times.

Recent articles proposed to improve and augment typical event-triggered state estimation (Battistelli et al. 2018, Huang et al. 2017, Shi et al. 2014) and control algorithms (Baumann et al. 2018, Narayanan & Jagannathan 2017, Vamvoudakis & Ferraz 2018, Vamvoudakis et al. 2019) with data-based techniques. In these works, learning is used to approximate intractable conditional probability densities that arise in distributed problems or to obtain tractable solutions to Hamilton-Jacobi-Bellman equations that yield optimal control policies, e.g., with model-free methods such as Q-learning, or based on neural networks based. However, none of these works considers principled decision making on model learning in order to improve prediction accuracy, as we do herein. Thus, we address the fundamentally different question of *when to learn*, leading to the new concept of ETL.

In order to obtain effective learning triggers, we take a probabilistic view on inter-communication times (i.e., the time between two communication events) and trigger learning experiments whenever the expected communication differs from the empirical. A similar probabilistic interpretation of inter-communication times is considered in (Xu & Hespanha 2004), where

NCSs are modeled as jump-diffusion processes, and the expected value of inter-communication times is considered. In this article, we analyze the statistical properties in a more general context and propose the design of learning triggers based on inter-communication times and concentration inequalities.

Adaptive control (Åström & Wittenmark 2013) aims to improve control performance by adapting model parameters continuously. Under certain persistent excitation conditions and for certain system classes, convergence to a reference trajectory and even parameter convergence can be guaranteed. However, divergence is a serious concern during nominal operation, where the persistent excitation conditions are not necessarily satisfied. This can often be a significant issue in practice. Here, we propose a different concept: rather than adapting or learning parameters all the time, we make a specific decision about *when to learn* a new model or parameters. By separating learning from nominal behavior, we ensure resource efficiency and informative data, which is collected in dedicated learning experiments.

Adaptive filtering, change, fault, and anomaly detection have been developed to identify defective and malfunctioning systems for a variety of applications (see (Gertler 2013, Gustafsson 2000, Isermann 1984, 2006) for an overview). In this article, we develop a method that is conceptually related to these research areas but closely tailored to NCSs with intermittent communication. In ETL, we combine tools from stochastic calculus and statistics to predict and analyze inter-communication times. Inter-communication times are independent, identically distributed, and scalar-valued. Due to these advantageous statistical properties, we can provide guarantees for the developed learning triggers.

Comparing expected behavior with observed realizations is also common in cognitive science to model human learning (Butz et al. 2003*a*). This effect is often quantified with the aid of internal models and anticipation along surprisal boundaries (Butz et al. 2003*b*). Interestingly, these ideas are very similar to ETL from an abstract point of view. However, the considered systems, concrete implementation, and developed theory differ significantly. Notwithstanding, there exists a conceptual connection to learning in biological systems.

Contributions

We propose the novel idea of event-triggered learning and develop the concept in the context of networked control systems. In particular, we derive data-driven learning triggers based on statistical properties of inter-communication times. These triggers make principled decisions on *when* to learn a new model.

In detail, this article makes the following contributions:

- introducing event-triggered learning with probabilistic guarantees ensuring the effectiveness of the proposed learning triggers;
- treatment of both cases, perfect state measurements and output measurements (Kalman filtering), and analysis of their differences; and
- demonstration of improved prediction accuracy and reduced communication in numerical simulations.

Preliminary results were presented in the conference paper (Solowjow et al. 2018). This article significantly extents this paper and presents the full ETL framework. While (Solowjow et al. 2018) considered only learning triggers based on the expected value and the perfect state measurement case, we generalize to improved triggers based on the full distribution and also extent to the more relevant case of output measurements (Kalman filtering). Furthermore, the theoretical properties of the sample-based triggers are improved, and new illustrative examples are presented.

2 Problem Formulation

In this section, we make the problem of event-triggered learning precise for linear Gaussian time-invariant systems. The framework is developed in the context of NCSs and primarily focuses on limited bandwidth. Information exchange over networks is abstracted to be ideal in the sense that there are no packet drops or delays. First, we state the problem formulation for continuous time systems. We then address the discrete time case separately since the technical details differ slightly. In Sec. 7.1, the problem is extended to output measurements and, in particular, the Kalman filter setting.

2.1 Continuous Time Formulation

Let $(S, \mathcal{F}, (\mathcal{F}_t)_{t\in\mathbb{R}^+}, \mathbb{P})$ be a filtered probability space and $X(t) \in \mathbb{R}^n$ a stochastic process, indexed by time $t \geq 0$. Furthermore, assume $X(t)$ (cf. 'Process' block in Fig. 1) is a solution to the following linear stochastic differential equation (SDE)

$$\mathrm{d}X(t) = \mathcal{A}X(t)\mathrm{d}t + \mathcal{Q}\,\mathrm{d}W(t), \quad X(0) = x_0. \tag{1}$$

Solutions to the SDE (1) are well investigated and also known as Ornstein-Uhlenbeck (OU) processes (Øksendal 2003). Further, let $\mathcal{A} \in \mathbb{R}^{n\times n}$ be a matrix with negative eigenvalues, which may in practice be obtained by applying local feedback control and considering the stable closed-loop dynamics. Assume $\mathcal{Q} \in \mathbb{R}^{n\times n}$ is a positive definite matrix, $W(t) \in \mathbb{R}^n$ a standard Wiener process that models process noise, and the initial point $x_0 \in \mathbb{R}^n$ is known. We denote the system parameters as $\theta = (\mathcal{A}, \mathcal{Q})$ and models as $\hat{\theta} = (\hat{\mathcal{A}}, \hat{\mathcal{Q}})$.

For the model-based predictions ('Model-based Predictions' in Fig. 1), we use the expected value of system (1), which coincides with the open-loop predictions of the deterministic system $\mathrm{d}\check{X}(t) = \hat{A}\check{X}(t)\mathrm{d}t$, with $\check{X}(0) = x_0$. Due to the stochasticity of the system, the prediction error will almost surely leave any predefined domain after sufficient time. Event-triggered communication ('State Trigger' in Fig. 1) bounds the prediction error by resetting the open-loop predictions $\check{X}(t)$ to the current state $X(t)$. Further, the binary event trigger

$$\gamma_{\text{state}} = 1 \iff \|X(t) - \check{X}(t)\|_2 \geq \delta, \tag{2}$$

is only activated when the error threshold $\delta > 0$ is crossed and hence, limits communication to necessary instances. The corresponding inter-communication time is defined as

$$\tau \coloneqq \inf\{t \in \mathbb{R} : \|X(t) - \check{X}(t)\|_2 \geq \delta\}, \tag{3}$$

and realizations of this random variable are denoted as $\tau_1, \ldots, \tau_n$ and can be directly measured as the time between communication instances.

Assumption 1 *We assume $\tau \leq \tau_{\max} < \infty$.*

Bounded communication times are usually implemented in real-world applications to detect defect agents, which never communicate. Hence, communication is enforced after $\tau_{\max}$. For the design of the final learning trigger to be derived in this article, the assumption can be omitted. However, it is useful for intermediate results, such as the expectation-based learning trigger.

We address the problem of designing learning triggers ('Learning Trigger' in Fig. 1) based on inter-communication time analysis. Since the probability distribution of τ can be fully parameterized by θ, we can derive an expected distribution based on the model $\hat{\theta}$ and test if empirical inter-communication times are drawn from that distribution. Further, this statistical analysis yields theoretical guarantees, which are obtained from concentration inequalities and ensure that the derived learning triggers are effective. Therefore, we design a method to perform dedicated learning experiments on necessity and update models $\hat{\theta}$ in an event-triggered fashion.

2.2 Discrete Time Formulation

Since processing on microcontrollers or sensors mostly happens on synchronously sampled data, we provide an alternative discrete time formulation of the considered problem. In principle, the problem formulation does not change. However, some essential details differ; for example, the inter-communication times from (3) need to be treated differently due to discontinuities in the states.

The discrete time analogue to (1) is

$$x(k+1) = Ax(k) + \epsilon(k), \quad x(0) = x_0, \tag{4}$$

with discrete time index $k \in \mathbb{N}$ and state $x(k) \in \mathbb{R}^n$. Furthermore, we assume $A \in \mathbb{R}^{n \times n}$ has all eigenvalues strictly within the unit sphere and $\epsilon(k) \sim \mathcal{N}(0, Q)$ with $Q \in \mathbb{R}^{n \times n}$ being symmetric and positive definite. The model-based predictions are obtained through the equation $\check{x}(k+1) = \hat{A}\check{x}(k)$, which yields the trigger

$$\gamma_{\text{state}} = 1 \iff \|x(k) - \check{x}(k)\|_2 \geq \delta. \tag{5}$$

We define the system parameters and model as $\theta = (A, Q)$ and $\hat{\theta} = (\hat{A}, \hat{Q})$. Hence, we obtain the inter-communication times

$$\tau^{\mathrm{d}} \coloneqq \min\{k \in \mathbb{N} : \|x(k) - \check{x}(k)\|_2 \geq \delta\}. \tag{6}$$

3 Communication as Stopping Times

In this section, we characterize inter-communication times (Eq. (3)) as stopping times of the prediction error process. The inter-communication time τ is a random variable and depends on the stochastic system (1). We seek to compare model-based expectations to observed data in order to detect significant inconsistencies between θ and $\hat{\theta}$. The core idea of the learning triggers comes down to deriving expected stopping time distributions based on the model $\hat{\theta}$ and then analyzing how likely it is that observed stopping times $\tau_1, \ldots, \tau_n$ are drawn from this distribution.

Assuming $\hat{\theta} = \theta$, we derive model-based statistical properties of τ. Later on, we will test the hypothesis that empirical inter-communication times are indeed drawn from the derived distribution of τ—if not, this will indicate $\hat{\theta} \neq \theta$; that is, the model does not match reality.

3.1 Theoretical Properties

We define the error process as $Z(t) \coloneqq X(t) - \check{X}(t)$. Due to linearity, it follows immediately that $Z(t)$ is an OU process as well and that $Z(0) = 0$. Next, we introduce inter-communication times with respect to the stochastic process $Z(t)$. Assume $\mathcal{F}_t = \sigma(Z_s : s \leq t)$ is the natural filtration on the given probability space and τ a stopping time with respect to $\mathcal{F}_t$. In particular, we consider the first exit time of the stochastic process $Z(t)$ from a sphere with radius δ, i.e., $\tau \coloneqq \inf\{t \in \mathbb{R} : \|Z(t)\|_2 > \delta\}$, which precisely coincides with (3). Hence, we use the terms stopping times and inter-communication times synonymously in this article.

After each communication instance, we reset the process $Z(t)$ and set it to zero again by correcting $\check{X}(t)$ to $X(t)$. The sample paths of the process $Z(t)$ are (almost surely) continuous between two inter-communication times, which follows from the existence and uniqueness theorem of solutions to SDEs (cf. (Øksendal 2003)).

Therefore, we can precisely quantify the moment when the error threshold (2) is crossed. Further, it is possible to quantify statistical properties of τ such as the expected value (Pavliotis 2014, Sec. 7.2) or the distribution (Patie & Winter 2008) with the aid of certain boundary value problems. In particular, there are existence and uniqueness theorems (Patie & Winter 2008) that imply that τ is mathematically well behaved.

3.2 Monte Carlo Approximations

Next, we describe how we obtain statistical properties of τ such as expected value $\mathbb{E}[\tau]$, variance $\mathbb{V}[\tau]$, and cumulative distribution function (CDF) $F(t)$ with the aid of sample-based methods and hence, without solving nonlinear boundary value problems. Given the system parameters θ, we can simulate trajectories of the stochastic process (1) with the aid of numerical sample methods such as the Euler-Maruyama scheme (cf. (Kloeden & Platen 2011, Sec. 10.2) for an introduction to numerical solutions of SDEs).

In order to obtain independent and identically distributed (i.i.d.) samples $\tau_1, \ldots, \tau_n$, we sample the process $Z(t)$ and restart from zero after reaching the threshold δ. Alternatively, we could also simulate $X(t)$ and $\check{X}(t)$ and set the predictions $\check{X}(\tau)$ to the true value $X(\tau)$ when communication is triggered. The statistical properties of the corresponding stopping times do not differ because the processes $Z(t)$ and $X(t) - \check{X}(t)$ are indistinguishable. Further, stable OU processes are stationary and satisfy the strong Markov property, which generalizes the Markov property to stopping times.

For given i.i.d. random variables, we can approximate the expected value with $\frac{1}{n}\sum_{i=1}^{n}\tau_i$ and the CDF with $F_n(t) := \frac{1}{n}\sum_{i=1}^{n}\mathbb{1}_{\tau_i \leq t}$, where $\mathbb{1}$ is the indicator function. Quantifying the convergence speed of the above approximation will be vital in designing learning triggers.

4 Learning Trigger Design for Continuous Time

In this section, we design the learning trigger γ_{learn} (cf. Fig. 1) to detect a mismatch between model and true dynamics based on the inter-communication time τ.

4.1 Concentration Inequalities

The following results will form the backbone of the later derived learning triggers. Concentration inequalities quantify the convergence speed of empirical distributions to their analytical counterparts. In particular, Hoeffding's inequality bounds the expected deviation between mean and expected value. Further, we also consider the Dvoretzky-Kiefer-Wolfowitz (DKW) inequality, which compares empirical and analytical CDF functions, and bounds the error between them uniformly. Essentially, we test if observed data fits the distribution, which is induced by the model $\hat{\theta}$; that is, which was derived with $\hat{\theta} = \theta$ (cf. Sec. 3). If the distributions do not match, we conclude an unfit model and update $\hat{\theta}$ through model learning.

Lemma 2 (Hoeffding (1963)) *Let $\tau_1, \ldots, \tau_n$ be i.i.d. bounded random variables, s. t. $\tau_i \in [0, \tau_{\max}]$. Then*

$$\mathbb{P}\left[\left|\frac{1}{n}\sum_{i=1}^{n}\tau_i - \mathbb{E}[\tau]\right| > \kappa\right] \leq 2\exp\left(-\frac{2n\kappa^2}{\tau_{\max}^2}\right). \tag{7}$$

We will first design learning triggers around the Hoeffding's inequality and later move on to richer statistical information. Therefore, we also want to analyze the convergence speed of the empirical CDF function.

Lemma 3 (DKW Inequality (Massart 1990)) *Assume $\tau_1, \ldots, \tau_n$ are i.i.d. random variables with CDF $F(t)$ and empirical CDF $F_n(t)$. Then*

$$\mathbb{P}\left[\sup_{t\in\mathbb{R}}|F_n(t) - F(t)| > \kappa\right] \leq 2\exp(-2n\kappa^2). \tag{8}$$

4.2 Expectation-based Learning Trigger

We propose a first learning trigger γ_{learn} based on the expected value $\mathbb{E}[\tau]$.

4.2.1 Exact Learning Trigger

Based on the foregoing discussion, we propose the following learning trigger:

$$\gamma_{\text{learn}} = 1 \iff \left|\frac{1}{n}\sum_{i=1}^{n}\tau_i - \mathbb{E}[\tau]\right| \geq \kappa_{\text{exact}}, \tag{9}$$

where $\gamma_{\text{learn}} = 1$ indicates that a new model shall be learned; $\mathbb{E}[\tau]$ is the analytical expected value, which is based on the model $\hat{\theta}$; and $\tau_1, \tau_2, \ldots, \tau_n$ are the last n empirically observed inter-communication times (τ_i the duration between two state triggers (2)). The horizon n is chosen to yield robust triggers in the sense that a larger time horizon allows the detection of smaller changes. However, it also increases the delay until the n samples are actually observed. The threshold parameter κ_{exact} quantifies the error we are willing to tolerate. There are some examples where it is possible to compute $\mathbb{E}[\tau]$ analytically. In general, however, it is intractable. Hence, we also propose the approximated learning trigger, which takes the approximations for the statistical analysis into account. We denote (9) as the *exact learning trigger* because it involves the exact expected value $\mathbb{E}[\tau]$, as opposed to the trigger derived in the next subsection, which is based on a Monte Carlo approximation of the expected value.

Even though the trigger (9) is meant to detect inaccurate models, there is always a chance that the trigger fires not due to an inaccurate model, but instead due to the randomness of the process (and thus randomness of inter-communication times τ_i). Such false positives are inevitable due to the stochastic nature of the problem. However, we obtain a confidence interval, which contains the empirical mean with high confidence. If observations violate the derived confidence interval, we conclude that distributions do not match, and learning is beneficial. Therefore, we propose to choose κ_{exact} to yield effective triggering with a user-defined confidence level. We then have the following result for the trigger (9):

Theorem 4 (Exact learning trigger) *Assume τ and $\tau_1, \dots, \tau_n$ are i.i.d. random variables and the parameters α, n, and $\tau_{\max}$ are given. If the trigger* (9) *gets activated* ($\gamma_{\text{learn}} = 1$) *with*

$$\kappa_{\text{exact}} = \tau_{\max}\sqrt{\frac{1}{2n}\ln\frac{2}{\alpha}}, \tag{10}$$

then

$$\mathbb{P}\left[\left|\frac{1}{n}\sum_{i=1}^{n}\tau_i - \mathbb{E}[\tau]\right| \geq \kappa_{\text{exact}}\right] \leq \alpha. \tag{11}$$

PROOF. Substituting κ_{exact} into the right-hand side of Hoeffding's inequality yields the desired result. □

The theorem quantifies the expected convergence rate of the empirical mean to the expected value for a perfect model. This result can be used as follows: the user specifies the desired confidence level α, the number n of inter-communication times considered in the empirical average, and the maximum inter-communication time $\tau_{\max}$. By choosing κ_{exact} as discussed, the exact learning trigger (9) is guaranteed to make incorrect triggering decisions (false positives) with a probability of less than α.

4.2.2 Approximated Learning Trigger

As discussed in Sec. 3, obtaining $\mathbb{E}[\tau]$ can be difficult and computationally expensive. Instead, we propose to approximate $\mathbb{E}[\tau]$ by sampling τ. For this, we simulate the process $Z(t)$ and average the obtained stopping times $\hat{\tau}_1, \dots, \hat{\tau}_m$. This yields the *approximated learning trigger*

$$\gamma_{\text{learn}} = 1 \iff \left|\frac{1}{n}\sum_{i=1}^{n}\tau_i - \frac{1}{m}\sum_{i=1}^{m}\hat{\tau}_i\right| \geq \kappa_{\text{approx}}. \tag{12}$$

The Monte Carlo approximation leads to a choice of κ_{approx}, which is different from κ_{exact} for small m. For $m \to \infty$ we see that κ_{approx} converges to κ_{exact}.

Theorem 5 (Approximated Learning Trigger) *Assume $\tau_1, \dots, \tau_n$, and $\hat{\tau}_1, \dots, \hat{\tau}_m$ are i.i.d. random variables. If the trigger* (12) *gets activated* ($\gamma_{\text{learn}} = 1$) *with*

$$\kappa_{\text{approx}} = \tau_{\max}\sqrt{\frac{n+m}{2nm}\ln\frac{2}{\alpha}}, \tag{13}$$

then

$$\mathbb{P}\left[\left|\frac{1}{n}\sum_{i=1}^{n}\tau_i - \frac{1}{m}\sum_{i=1}^{m}\hat{\tau}_i\right| \geq \kappa_{\text{approx}}\right] \leq \alpha. \tag{14}$$

PROOF. First, we introduce an alternative formulation of Hoeffding's inequality (7)

$$\mathbb{P}\left[\left|\frac{1}{n}\sum_{i=1}^{n}\tau_i - \frac{1}{m}\sum_{i=1}^{m}\hat{\tau}_i - (\mathbb{E}[\tau] - \mathbb{E}[\hat{\tau}])\right| > \kappa_{\text{approx}}\right] \\ \leq 2\exp\left(-\frac{2\kappa_{\text{approx}}^2}{(m^{-1}+n^{-1})\tau_{\max}^2}\right),$$

which was already stated in the original article by Hoeffding (Hoeffding 1963) as a corollary (Eq. 2.6). Here, we assume that τ and $\hat{\tau}$ are identically distributed and, therefore, the analytical expected values cancel out. Rearranging for κ_{approx} yields the desired result. □

4.3 Density-based Learning Trigger

Analyzing the expected values is, in general, not enough to distinguish random variables since higher moments such as variance can differ. Therefore, we propose to look at the CDF, build learning triggers around the DKW inequality (8), and thus use richer statistical information. We propose the following learning trigger:

$$\gamma_{\text{learn}} = 1 \iff \sup_{t\in\mathbb{R}}|F(t) - F_n(t)| > \kappa_{\text{exact}}. \tag{15}$$

The density-based learning trigger has the following property:

Theorem 6 (Exact Density Learning Trigger) *Assume $\tau_1, \dots, \tau_n$ are i.i.d. random variables with CDF $F(t)$ and empirical CDF $F_n(t)$. If the learning trigger* (15) *gets activated* ($\gamma_{\text{learn}} = 1$) *with*

$$\kappa_{\text{exact}} = \sqrt{\frac{1}{2n}\ln\frac{2}{\alpha}}, \tag{16}$$

then

$$\mathbb{P}\left[\sup_{t\in\mathbb{R}} |F(t) - F_n(t)| > \kappa_{\text{exact}}\right] \leq \alpha. \tag{17}$$

PROOF. Follows directly from the DKW Inequality. □

Finally, we can follow the reasoning as before and obtain the sample-based version of the trigger (15)

$$\gamma_{\text{learn}} = 1 \iff \sup_{t\in\mathbb{R}} |\hat{F}_m(t) - F_n(t)| > \kappa_{\text{approx}}, \tag{18}$$

where $\hat{F}_m(t) = \frac{1}{m}\sum_{i=1}^{m} \mathbb{1}_{\hat{\tau}_i \leq t}$ and $\hat{\tau}_i$ are obtained by creating samples based on the model $\hat{\theta}$. This trigger is essentially the well established two-sample Kolmogorov-Smirnov (KS) test (Hodges 1958).

Theorem 7 (Two-sample KS Learning Trigger) *Assume $\tau_1, \ldots, \tau_n$ and $\hat{\tau}_1, \ldots, \hat{\tau}_m$ are i.i.d. random variables with empirical CDFs $F_n(t)$ and $\hat{F}_m(t)$. If the trigger* (18) *gets activated with*

$$\kappa_{\text{approx}} = \sqrt{\frac{n+m}{2nm}\ln\left(\frac{2}{\alpha}\right)}, \tag{19}$$

then

$$\mathbb{P}\left[\sup_{t\in\mathbb{R}} |\hat{F}_m(t) - F_n(t)| > \kappa_{\text{approx}}\right] \leq \alpha. \tag{20}$$

PROOF. Follows from the two-sample KS test. □

The density-based learning triggers do not depend on $\tau_{\max}$ and consider richer statistical information, which can be an advantage and will be discussed in detail in the experimental sections.

5 Learning Trigger Design for Discrete Time

Based on the previous discussion, we will now highlight how to apply the derived learning triggers to discrete time systems (4). The random variables τ (cf. Eq. (3)) and τ^{d} (cf. Eq. (6)) can differ significantly due to discretization effects. Intuitively, this effect can be thought of as the continuous time process crossing the δ-threshold and returning within the discretization time. Therefore, the discrete-time process has no possibility of observing the crossing, and hence, stopping times tend to be larger for discrete time systems. For small time steps, the difference tends to be negligible, and τ^{d} converges to τ in the limit.

In this section, we show that the *approximated* learning triggers transfer without any modification to the discrete time system. It is important to adjust the system parameters $\theta = (A, Q)$ and the model $\hat{\theta} = (\hat{A}, \hat{Q})$ to discrete time (cf. Sec. 2.2) in order to sample from the correct distribution (i.e., sampling from the continuous time model, while the true dynamics are discrete, or vice versa). Only based on statistical tests, irrelevant of the actual shape, we decide if they coincide.

Theorem 8 (Discrete Time Learning Trigger) *Assume θ and $\hat{\theta}$ correspond to the discrete time system* (4)*. Then, the previously derived approximated learning triggers* (12) *and* (18) *are applicable without any further modification.*

PROOF. The derived learning triggers test if given observations of inter-communication times $\tau_1, \ldots, \tau_n$ are drawn from the same distribution as $\hat{\tau}_1, \ldots, \hat{\tau}_m$ and therefore, if $\theta = \hat{\theta}$. The concrete shape of the distribution is irrelevant for the test. □

Remark 9 *It is also possible to consider more complex noise models such as colored noise. The main challenge lies in identifying the system and, in particular, the noise model from data.*

6 Numerical Example – Reduced Communication

The learning triggers derived in the previous two sections are the core element in the proposed ETL architecture (Fig. 1, block 'Learning Trigger'). For 'Model Learning' in the context of linear Gaussian systems considered herein, one can use standard techniques for linear systems identification (Ljung 1999), which we do not elaborate further. Thus, all components of the proposed ETL method in Fig. 1 are complete, and we present a first numerical example to illustrate the main ideas of the developed learning triggers.

6.1 Setup

Next, we introduce the system, and afterward, we apply the learning trigger in order to demonstrate how to detect an inaccurate model. We consider the first-order dynamical system $x(k+1) = 0.9x(k) + \epsilon(k)$, with noise $\epsilon(k) \sim \mathcal{N}(0, 1)$. Further, we assume the disturbed model $\hat{\theta} = (0.8, 1)$ and hence, we obtain the predictions with the equation $\check{x}(k+1) = 0.8\check{x}(k)$. To bound the prediction error, we deploy the state trigger (5) with $\delta = 3$. In Fig. 2, we can see in the first graph a trajectory of states $x(k)$ as a black dashed line and the model-based predictions $\check{x}(k)$ in blue. Whenever $\gamma_{\text{state}} = 1$, we set $\check{x}(\tau)$ to $x(\tau)$. The error signal never crosses the δ-threshold

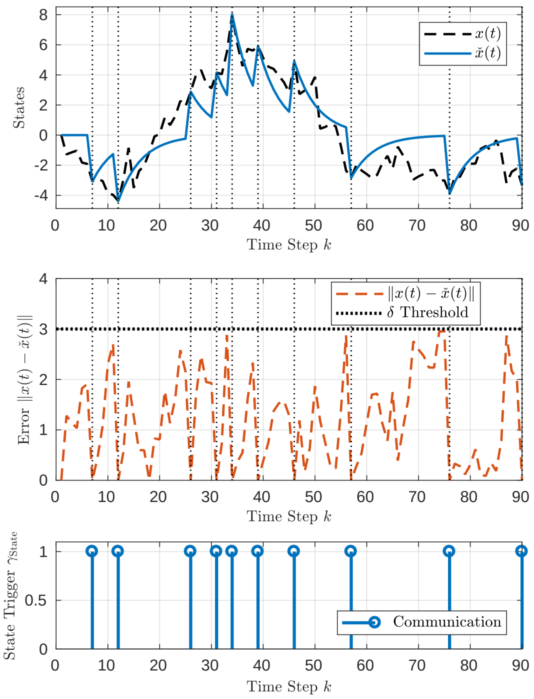


Fig 2. Example of model-based state predictions, which are reset to the exact state whenever the error would exceed the predefined threshold $\delta = 3$. States, error signals, and the first ten communication instances τ_i are depicted in the three graphs (top to bottom).

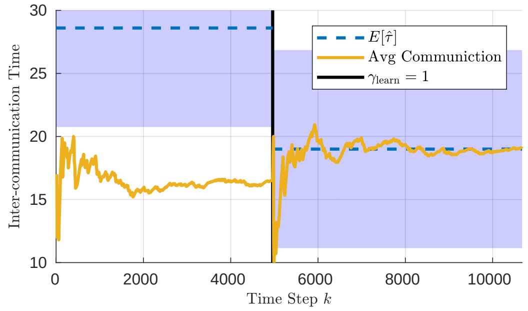


Fig 3. Average inter-communication times. The dashed blue line illustrates the model-based expected value and the yellow line the empirical mean. The shaded blue area around the dashed blue line indicates the confidence interval that should contain the empirical mean with 95% probability. Every new communication instant is added to a buffer of size $n = 300$. Afterward, the learning trigger compares the expected value with empirical mean and updates the model if the yellow line is outside the blue area. A new model-based expected value is computed, and the empirical mean coincides with it. Further, the inter-communication times increase, which corresponds to a decrease in communication.

and is depicted in the second graph. The communication instances are shown in the third graph. The distances between two consecutive communication instances corresponds to the inter-communication times. Further, we set $\alpha = 0.05$, $\tau_{\max} = 100$, $n = 300$, and $m = 100\,000$.

6.2 Expectation-based Learning Trigger

The proposed learning triggers analyze the statistical properties of the observed inter-communication times and compare them to model-induced quantities. For instance, in Fig. 3, the corresponding average inter-communication rate is shown (yellow line). The first n inter-communication times are stored in a buffer, and the empirical mean is computed. Based on the model $\hat{\theta}$ and with the aid of m Monte Carlo simulations, we derive $\mathbb{E}[\hat{\tau}] \approx 28.6$ (dashed blue line). The model-based confidence interval is obtained with the aid of Theorem 5. After the buffer is successfully filled (at $k = 4961$), the learning trigger (12) compares if the buffered average inter-communication rate (yellow line) lies outside the expected confidence interval, which is the case here. Therefore, the learning trigger discovers an inaccurate model and triggers a learning experiment. Here, we abstract learning and set model $\hat{\theta}$ to the true parameters θ. A more detailed discussion on the learning aspect of ETL can be found in (Solowjow et al. 2018), where we demonstrated the effectiveness of ETL in hardware experiments on a cart-pole system.

After updating the model (at $k = 4961$), we empty the buffer, start collecting new stopping times, and reset the average inter-communication time accordingly. This causes the initial fluctuation of the signal. However, we see fast convergence to the model-based expectation. Further, the average inter-communication time was increased after updating the model (yellow line converges to a larger value), which results in less communication.

The test statistic is also depicted in Fig. 4 for the initial inaccurate model and in Fig. 5 for the exact model. The dashed blue line again represents the model-based expected value, while the dashed red line depicts the empirical mean at the moment of triggering.

6.3 Density-based Learning Triggers

Next, we will discuss the density-based learning trigger (18), which is also illustrated in Fig. 4 and Fig. 5. The solid blue line represents the model-based CDF function $\hat{F}_m(t)$, and the solid red line is the empirical CDF $F_n(t)$ based on observed inter-communication times. Here, both triggers detect the inaccurate model (cf. Fig. 4) and have high confidence in the true model since the model-based and empirical quantities coincide, which is depicted in Fig. 5. The confidence interval is derived with Theorem 7 and tighter than the expectation-based. Hence, inaccurate models can be detected faster.

6.4 Expected Value is not Enough

Here, we assume the model $\hat{\theta} = (0.5, 1.7)$ with $n = m = 10\,000$. Intuitively, process noise and stability have op-

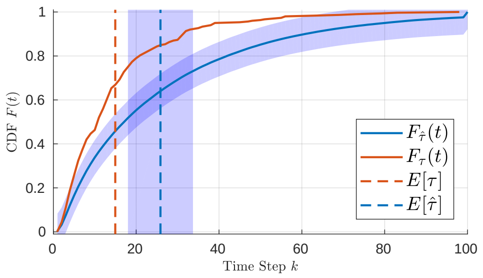


Fig 4. Statistical properties of expected communication $\hat{\tau}$ (blue) and observed inter-communication times τ (red). The dashed lines capture the expected values, and the shaded blue region is a confidence interval that should contain the dashed red line with a 95% probability (cf. (12)). As there is a significant deviation between observation and expectation, the learning trigger initiates to relearn the model. The solid lines represent the CDF functions, and also here, the empirical distribution is not contained within the confidence bounds, which triggers learning (cf. (18)).

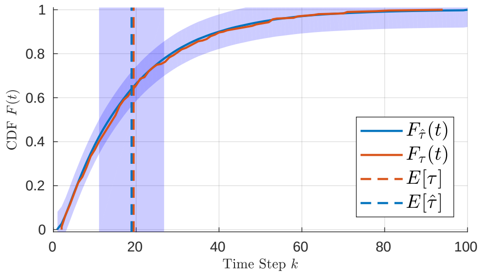


Fig 5. After updating the model $\hat{\theta}$ to the true system parameters θ, the estimated stopping times $\hat{\tau}$ coincide with empirical stopping times τ. As a direct consequence, communication behavior is improved as the stopping times increase.

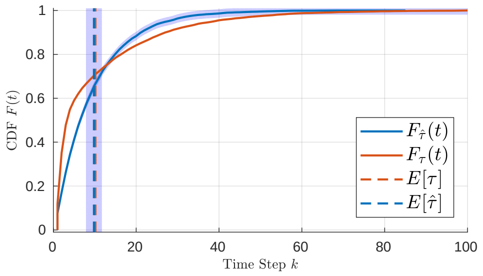


Fig 6. Counterexample showing that the expected value of inter–communication times (learning trigger (12)) may not be sufficient to detect an inaccurate model. Furthermore, the CDF-based trigger (18) detects the mismatch reliably.

posite effects on the communication behavior. We construct the counterexamples by creating a hypersurface of models $\hat{\theta}$, where the noise and stability effects cancel out. Figure 6 shows the expected and empirical expected values of inter-communication times, which are almost identical – bad performance is expected and also realized due to the bad prediction model. The CDF-based trigger is still able to detect the inaccuracy, which is a big advantage over the expectation-based learning triggers. Clearly, the highest inter-communication time is realized when $\hat{\theta} = \theta$, which can be observed when comparing Fig. 5 with Fig. 6, where the inter-communication time is twice as high.

Clearly, the model has also to be communicated at some point. However, this happens very rarely and, in particular, when there is a significant change in the system. We conclude that both learning triggers are effective in detecting a mismatch between model and true dynamics. Also, average communication was successfully reduced after updating the model.

7 Extensions and Insights

So far, we assumed that perfect measures of the full state $x(k)$ are available at the sending agent. In the following, we will drop this assumption and consider systems where only part of the state can be measured.

7.1 Output Measurements

Assume the following system

$$x(k+1) = Ax(k) + \epsilon(k), \quad y(k) = Cx(k) + \nu(k), \tag{21}$$

with output measurements $y(k) \in \mathbb{R}^m$. Further, let $A \in \mathbb{R}^{n \times n}$ and $C \in \mathbb{R}^{n \times m}$. The system is again assumed to be stochastic with process noise $\epsilon(k) \sim \mathcal{N}(0, Q)$ and observation noise $\nu(k) \sim \mathcal{N}(0, R)$, which are independent of each other. We also assume that A is stable, and the pair (A, C) is observable. Hence, the system is parameterized by $\theta = (A, C, Q, R)$ and modeled by $\hat{\theta} = (\hat{A}, \hat{C}, \hat{Q}, \hat{R})$. To reconstruct the full state, we use a Kalman filter (KF), which is the optimal filter for linear Gaussian systems with exact models (Anderson & Moore 2012). Here, we consider the steady-state KF and obtain

$$\hat{x}(k+1) = \hat{A}\hat{x}(k) + K\left(y(k+1) - \hat{C}\hat{A}\hat{x}(k)\right), \tag{22}$$

where $K \in \mathbb{R}^{m \times n}$ is the corresponding gain. Further, assume the process has already converged to stationarity.

Ideally, we want to use the KF states $\hat{x}(k)$ on the receiving agent's site. However, this would require periodic communication of the estimates $\hat{x}(k)$, or the measurements $y(k)$, which we try to avoid. Exactly as in Sec. 2, we run a model-based prediction step in the absence of data, obtain $\check{x}(k+1) = \hat{A}\check{x}(k)$, and employ the state trigger $\|\hat{x}(k) - \check{x}(k)\|_2 \geq \delta$. Hence, the inter-communication time is defined as $\tau^{\mathrm{o}} := \min\{k \in \mathbb{N} : \|\hat{x}(k) - \check{x}(k)\|_2 \geq \delta\}$. Extending ETL to output measurements is based on treating the KF sequence as a stochastic process in its own right and investigating the distribution of $\hat{x}(k)$. With the aid of the innovation sequence, we can derive an auto-regressive structure. The

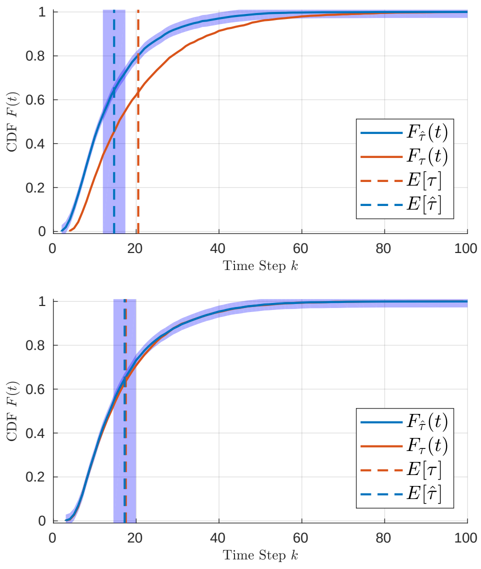


Fig 7. Communication behavior of a system with output measurements. In the first graph, we see the test statistic for an inaccurate model and in the second for $\hat{\theta} = \theta$. Both learning triggers ((12) and (18)) are effective in detecting the model mismatch. Interestingly, updating the model results in more communication, as can be seen by a decrease in the actual average inter-communication time $E[\tau]$ (dashed red) between top and bottom. With the improved model, the KF tracks the true states better, and thus, we obtain more communication.

innovation of the KF is defined as $I(k) = y(k) - C\hat{x}(k)$. Furthermore, it is well known that $I(1), \ldots, I(n)$ are independent normal distributed random variables with $I(k) \sim \mathcal{N}(0, S)$ (Anderson & Moore 2012). The covariance is given by $S = CPC^\mathsf{T} + R$, where P is the stationary error covariance matrix of the KF and can be obtained by solving the corresponding Riccati equation (Anderson & Moore 2012, Equation (4.4)). Hence, we reformulate the KF as

$$\hat{x}(k+1) = A\hat{x}(k) + KI(k), \tag{23}$$

and regard $I(k) \sim \mathcal{N}(0, S)$ as a random variable. By regarding $KI(k)$ as process noise, we are back to the previously discussed problem (cf. (4)) and can apply the derived tools and learning triggers. Hence, we can effectively analyze the distribution of the corresponding stopping time with the previously derived tools – sampling (23) to obtain model-based stopping times $\hat{\tau}^\mathrm{o}$.

7.2 *Better Models may Result in more Communication*

More accurate models result in better predictions. Thus, one may expect that improved models also lead to reduced communication of state information from sender to receiver (cf. Fig. 1). While this is indeed the case for perfect state measurements (as has been observed in the example of Sec. 6), it may actually be the opposite for the KF setting. Here, we present an example that demonstrates this rather unexpected effect – better models may lead to more communication. The reason is as follows: better models increase the KF performance, and thus, it is possible to track the unobserved states better. Therefore, it is possible to construct examples where communication increases, which is desirable for performance, though counterintuitive. Consider system (21) with the matrices

$$A = \begin{pmatrix} 1.000 & 0.010 & -0.005 & 0.000 \\ 0.017 & 1.027 & -0.301 & -0.061 \\ 0.000 & 0.000 & 0.997 & 0.009 \\ 0.046 & 0.067 & -0.507 & 0.850 \end{pmatrix}, C^\mathsf{T} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix} \tag{24}$$

which is obtained by linearizing the closed-loop dynamics of a stabilized inverted pendulum. We assume process noise $\epsilon(k) \sim \mathcal{N}(0, 0.1I_4)$ and observation $\nu(k) \sim \mathcal{N}(0, 0.1I_2)$, where I_n is the identity matrix of dimension n. Further, we assume that $\hat{\nu}(k) \sim \mathcal{N}(0, 0.5I_2)$ and that the model otherwise coincides with the true system parameters. We consider the KF states $\hat{x}(k)$ (cf. (22)), the predictions $\check{x}(k)$, and the state trigger

$$\gamma_\mathrm{state} = 1 \iff \|\hat{x}(k) - \check{x}(k)\|_2 \geq 1. \tag{25}$$

We initialize $x(0) = \hat{x}(0) = \check{x}(0) = 0$ and obtain the distribution over stopping times depicted in Fig. 7. The expected model-based communication is derived via Monte Carlo simulation of the innovation process (23), where we set $\tau_\mathrm{max} = 100$ and $m = n = 5000$.

In the first graph in Fig. 7, we can see that the empirical inter-communication times are higher than the model-based. Updating the model actually reduces the average inter-communication time (more communication), which is because the KF improves and tracks the states $x(k)$ better, which is illustrated in Fig. 8. The first plot shows the tracking performance when a perfect model is used $\hat{\theta} = \theta$ (KF states in yellow). For the second plot, we changed the covariance of $\nu(k)$ to $\mathcal{N}(0, 0.5I_2)$, as discussed above. In the third plot, we exaggerated this effect even further by assuming $\nu(k) \sim \mathcal{N}(0, 10I_2)$ in the model. In all three plots, we consider the first $k = 150$ time steps and stop afterward. In the third graph, we can see that the KF states deviate a lot from the true underlying states. However, they are still close to the open-loop predictions, and hence, there is very little communication. Despite the counterintuitive link between model accuracy and average communication, the example shows that the derived learning triggers are effective in detecting model mismatch.

8 Discussion and Future Work

Event-triggered learning is proposed in this article as a novel concept to trigger model learning when needed.

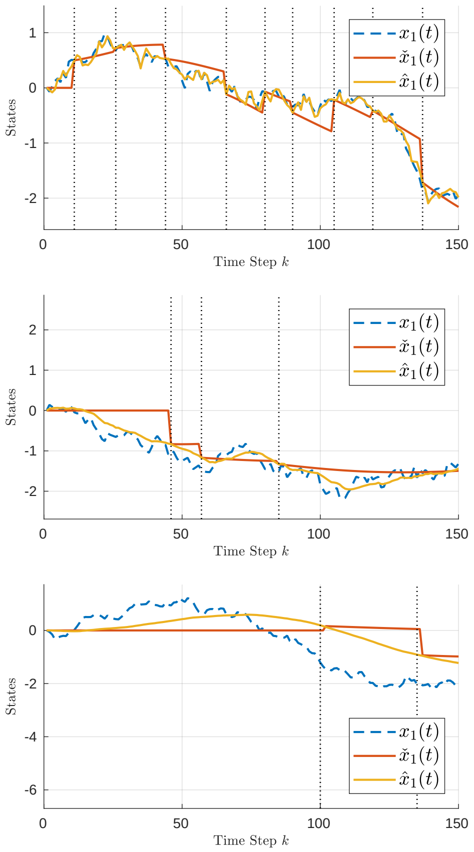


Fig 8. State trajectories of the first dimension of the four-dimensional system with output measurements (24). The state trigger (25) is applied and therefore, communication triggered when $\hat{x}(k)$ and $\check{x}(k)$ deviate by δ. Communication instances are depicted with dotted vertical lines. In the first graph, we can see how well the KF $\hat{x}(k)$ (yellow) tracks the true states $x(k)$ (dashed blue). The red line depicts the open-loop predictions $\check{x}(k)$. By worsening the model from the first to the second graph, the KF performance gets worse, which results in less communication. From the second to the third graph, we worsen the model even more, which results in even less communication, because the KF is not able to track the states.

This article focuses on the rigorous design of learning triggers, and we obtained (provably) effective learning triggers utilizing statistical tests. The concept of ETL has also already been applied in hardware experiments (Solowjow et al. 2018) and shown to yield reduced communication.

While event-triggered learning has been motivated as an extension to already existing methods to reduce communication in NCSs, the concept generally addresses the fundamental question of *when to learn* and potentially has much broader relevance. Here, we ultimately care about communication, and thus, it is a natural idea to analyze inter-communication times and trigger model learning based on these. Furthermore, they show advantageous statistical properties, such as being i.i.d. and scalar-valued. Depending on the concrete problem at hand, the signal used for triggering learning should be chosen accordingly. Using control performance as such a triggering signal is a potential extension, and first steps in this direction were taken in (Baumann et al. 2019).

References

Anderson, B. D. & Moore, J. B. (2012), *Optimal filtering*, Courier Corporation.

Åström, K. J. & Wittenmark, B. (2013), *Adaptive control*, Courier Corporation.

Battistelli, G., Chisci, L. & Selvi, D. (2018), 'A distributed kalman filter with event-triggered communication and guaranteed stability', *Automatica* **93**, 75–82.

Baumann, D., Solowjow, F., Johansson, K. H. & Trimpe, S. (2019), Event-triggered pulse control with model learning (if necessary), *in* 'Proceedings of the American Control Conference', pp. 792–797.

Baumann, D., Zhu, J.-J., Martius, G. & Trimpe, S. (2018), Deep reinforcement learning for event-triggered control, *in* 'Proceedings of the 57th IEEE International Conference on Decision and Control', pp. 943–950.

Butz, M. V., Sigaud, O. & Gérard, P. (2003*a*), Anticipatory behavior: Exploiting knowledge about the future to improve current behavior, *in* 'Anticipatory behavior in adaptive learning systems', Springer, pp. 1–10.

Butz, M. V., Sigaud, O. & Gérard, P. (2003*b*), Internal models and anticipations in adaptive learning systems, *in* 'Anticipatory behavior in adaptive learning systems', Springer, pp. 86–109.

Gertler, J. (2013), *Fault detection and diagnosis*, Springer.

Gustafsson, F. (2000), *Adaptive filtering and change detection*, Vol. 1, Citeseer.

Han, D., Mo, Y., Wu, J., Weerakkody, S., Sinopoli, B. & Shi, L. (2015), 'Stochastic event-triggered sensor schedule for remote state estimation', *IEEE Transactions on Automatic Control* **60**(10), 2661–2675.

Heemels, M., Johansson, K. H. & Tabuada, P. (2012), An introduction to event-triggered and self-triggered control, *in* 'Proceedings of the 51st IEEE International Conference on Decision and Control', IEEE, pp. 3270–3285.

Hespanha, J. P., Naghshtabrizi, P. & Xu, Y. (2007), 'A survey of recent results in networked control systems', *Proc. of IEEE* Special Issue on Technology of Networked Control Systems **95**(1), 138–162.

Hodges, J. (1958), 'The significance probability of the smirnov two-sample test', *Arkiv för Matematik* **3**, 469–486.

Hoeffding, W. (1963), 'Probability inequalities for sums of bounded random variables', *Journal of the American statistical association* **58**(301), 13–30.

Huang, J., Shi, D. & Chen, T. (2017), 'Energy-based event-triggered state estimation for hidden markov models', *Automatica* **79**, 256–264.

Isermann, R. (1984), 'Process fault detection based on modeling and estimation methods—a survey', *Automatica* **20**(4), 387–404.

Isermann, R. (2006), *Fault-diagnosis systems: an introduction from fault detection to fault tolerance*, Springer Science & Business Media.

Kloeden, P. & Platen, E. (2011), *Numerical Solution of Stochastic Differential Equations*, Stochastic Modelling and Applied Probability, Springer Berlin Heidelberg.

Lemmon, M. (2010), 'Event-triggered feedback in control, estimation, and optimization', *Networked Control Systems* **406**, 293–358.

Ljung, L. (1999), *System identification: Theory for the User*, Prentice Hall, New Jersey.

Massart, P. (1990), 'The tight constant in the dvoretzky-kiefer-wolfowitz inequality', *The annals of Probability* pp. 1269–1283.

Miskowicz, M. (2006), 'Send-on-delta concept: an event-based data reporting strategy', *Sensors* **6**(1), 49–63.

Miskowicz, M. (2015), *Event-based control and signal processing*, CRC Press.

Narayanan, V. & Jagannathan, S. (2017), 'Event-triggered distributed control of nonlinear interconnected systems using online reinforcement learning with exploration', *IEEE transactions on cybernetics* **48**, 2510–2519.

Øksendal, B. (2003), Stochastic differential equations, *in* 'Stochastic differential equations', Springer.

Patie, P. & Winter, C. (2008), 'First exit time probability for multidimensional diffusions: a pde-based approach', *Journal of computational and applied mathematics* **222**(1), 42–53.

Pavliotis, G. (2014), *Stochastic Processes and Applications Diffusion Processes, the Fokker-Planck and Langevin Equations*, Springer.

Shi, D., Chen, T. & Shi, L. (2014), 'Event-triggered maximum likelihood state estimation', *Automatica* **50**(1), 247–254.

Shi, D., Shi, L. & Chen, T. (2015), *Event-Based State Estimation: A Stochastic Perspective*, Springer.

Sijs, J., Kester, L. & Noack, B. (2014), A study on event triggering criteria for estimation, *in* '17th International Conference on Information Fusion', pp. 1–8.

Sijs, J. & Lazar, M. (2012), 'Event based state estimation with time synchronous updates', *IEEE Transactions on Automatic Control* **57**(10), 2650–2655.

Solowjow, F., Baumann, D., Garcke, J. & Trimpe, S. (2018), Event-triggered learning for resource-efficient networked control, *in* 'Proceedings of the American Control Conference'.

Suh, Y. S. (2007), 'Send-on-delta sensor data transmission with a linear predictor', *Sensors* **7**(4), 537–547.

Trimpe, S. (2017), 'Event-based state estimation: An emulation-based approach', *IET Control Theory & Applications* **11**(11), 1684–1693.

Trimpe, S. & Baumann, D. (2019), 'Resource-aware IoT control: Saving communication through predictive triggering', *IEEE Internet of Things Journal* .

Trimpe, S. & Campi, M. C. (2015), On the choice of the event trigger in event-based estimation, *in* 'Int. Conf. on Event-based Control, Communication, and Signal Processing', pp. 1–8.

Trimpe, S. & D'Andrea, R. (2011), 'An experimental demonstration of a distributed and event-based state estimation algorithm', *IFAC Proceedings Volumes* **44**(1), 8811–8818.

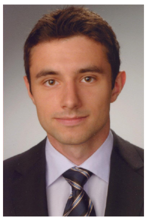
Trimpe, S. & D'Andrea, R. (2014), 'Event-based state estimation with variance-based triggering', *IEEE Transactions on Automatic Control* **59**(12), 3266–3281.

Vamvoudakis, K. G. & Ferraz, H. (2018), 'Model-free event-triggered control algorithm for continuous-time linear systems with optimal performance', *Automatica* **87**, 412–420.

Vamvoudakis, K. G., Pour Safaei, F. R. & Hespanha, J. P. (2019), 'Robust event-triggered output feedback learning algorithm for voltage source inverters with unknown load and parameter variations', *International Journal of Robust and Nonlinear Control* **29**(11), 3502–3517.

Wu, J., Jia, Q.-S., Johansson, K. H. & Shi, L. (2013), 'Event-based sensor data scheduling: Trade-off between communication rate and estimation quality', *IEEE Transactions on Automatic Control* **58**(4), 1041–1046.

Xu, Y. & Hespanha, J. P. (2004), Communication logics for networked control systems, *in* 'Proceedings of the American Control Conference', pp. 572–577.



Friedrich Solowjow received B. Sc. degrees in Mathematics and Economics from the University of Bonn in 2014 and 2015, respectively, and a M. Sc. degree in Mathematics from the University of Bonn in 2017. He is a PhD student in the Intelligent Control Systems Group at the Max Planck Institute for Intelligent Systems, Stuttgart, Germany and a member of the International Max Planck Research School for Intelligent Systems.

Sebastian Trimpe (M'12) received the B. Sc. degree in general engineering science and the M.Sc. degree (Dipl.-Ing.) in electrical engineering from Hamburg University of Technology, Hamburg, Germany, in 2005 and 2007, respectively, and the Ph. D. degree (Dr. sc.) in mechanical engineering from ETH Zurich, Zurich, Switzerland, in 2013. He is currently a Research Group Leader at the Max Planck Institute for Intelligent Systems, Stuttgart, Germany, where he leads the independent Max Planck Research Group on Intelligent Control Systems. His main research interests are in systems and control theory,machine learning, networked and autonomous systems. Dr. Trimpe is recipient of several awards, among others, the triennial IFAC World Congress Interactive Paper Prize (2011), the Klaus Tschira Award for achievements in public understanding of science (2014), the Best Demo Award of the International Conference on Information Processing in Sensor Networks (2019), and the Best Paper Award of the International Conference on Cyber-Physical Systems (2019).