

Efficient Encoding of Dynamical Systems through Local Approximations

Friedrich Solowjow^{1,*}, Arash Mehrjou^{2,*}, Bernhard Schölkopf², Sebastian Trimpe¹

Abstract—An efficient representation of observed data has many benefits in various domains of engineering and science. Representing static data sets, such as images, is a living branch in machine learning and eases downstream tasks, such as classification, regression, or decision making. However, the representation of dynamical systems has received less attention. In this work, we develop a method to represent a dynamical system efficiently as a combination of a state and a local model, which fulfills a criterion inspired by the minimum description length (MDL) principle. The MDL principle is used in machine learning and statistics to quantify the trade-off between the ability to explain seen data and the model complexity. Networked control systems are a prominent example, where such a representation is beneficial. When many agents share a network, information exchange is costly and should thus happen only when necessary. We empirically show the efficiency of the proposed encoding for several dynamical systems and demonstrate reduced communication for event-triggered state estimation problems.

I. INTRODUCTION

Nonlinear functions are essential for modeling complex physical systems. However, utilizing nonlinear models for practical applications is difficult and computationally expensive. Approximations are often required in order to obtain feasible algorithms. In this paper, we propose a framework that adaptively chooses *simple* local models, which are chosen as complicated as required to obtain a desired approximation accuracy. The intuition behind our approach goes back to Occam’s razor principle, which states that among competing models with equal ability to explain seen data, the simplest model should be preferred. Based on Occam’s razor principle, rigorous methods in machine learning have been established [1]. One of them is the minimum description length (MDL) principle, which is frequently used to investigate the connection between compression and generalization, cf. [2]. The main idea of the MDL principle comes down to describing a given data set with the aid of a hypothesis or law. This way, the information content of the data set can be compressed. However, in order to achieve a benefit in terms of compression, the hypothesis also needs to be simple enough with respect to the data set. Therefore, the MDL-principle quantifies the trade-off

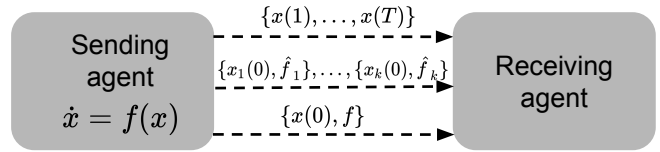


Fig. 1. Illustration of different encoding schemes of dynamical systems. Depicted are two agents, where one receiving agent requires the states of a sending agent. There are many possibilities how to transmit information about the dynamical system. This can be state data itself $x(1), \dots, x(T)$, or an initial point $x(0)$ and the dynamical function f , based on which it is possible to generate the whole trajectory of states. Since the dynamical function f is an infinite dimensional object in general, it can be infeasible to transmit it. Therefore, we develop a scheme that adaptively chooses local dynamics approximations \hat{f} , and propose this as an efficient encoding of dynamical system information.

between accuracy and complexity of a model and yields an efficient representation of data. Such an efficient encoding of dynamical systems has applications in various areas, such as statistical inference [3], compression, and transmission [4]—basically, whenever dynamics or state information need to be transmitted or stored efficiently.

Event-triggered state estimation (ETSE) has been developed over the last decade [5]–[10] for efficient transmission of data in networked control systems. In common ETSE approaches, dynamical system models are used to predict agents’ behavior. This way, periodic transmission of state data can be avoided, and state updates are only transmitted whenever the model predictions are not sufficient. This stream of work has recently been extended in [11] to also consider communication of dynamical system models, in addition to states. However, the framework in [11] has been developed for linear systems and the extension to nonlinear systems is nontrivial. This is due to the fact that the function f , describing dynamical system $\dot{x}(t) = f(x(t))$, is an infinite dimensional object and that it is not clear how to represent the information content of this system in general. We assume that data from this system is available as a time-sampled signal, as is usually the case in practice. Therefore, there are two extremes in representing the history of the states $x(t)$ over a given time horizon T . One is storing or transmitting the states $x(t)$ for each time instance. The other one would be storing an initial point x_0 and the dynamical function f . For a deterministic system, the former can be produced by the latter, however the required storage space will be very different if the function f can be represented efficiently. In this paper, we investigate an optimal middle point within this spectrum by finding a tractable representation of f . These different possibilities are depicted in Fig. 1, where two agents

*Equally contributing.

¹Intelligent Control Systems Group, Max Planck Inst. for Intelligent Systems, 70569 Stuttgart, Germany. Email: {fsolowjow, trimpe}@is.mpg.de.

²Empirical Inference Department, Max Planck Inst. for Intelligent Systems, 72076 Tübingen, Germany. Email: {amehrjou, bs}@tue.mpg.de.

This work was supported in part by the Max Planck Society, the Cyber Valley Initiative, and the German Research Foundation (DFG) grant TR 1433/1-1. The authors thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting Friedrich Solowjow.

that exchange information are illustrated. Furthermore, the solution in between is also proposed, i.e., local approximations f_1, \dots, f_k .

For the scenario, where two agents transmit information as in Fig. 1, we seek to minimize overall communication between agents by transmitting local models in addition to state data. The MDL principle provides a structured approach to achieve this and yields an unifying framework for discussing communication of state and model data in the context of ETSE. With the aid of the proposed algorithm, efficient and lighter models can be found that outperform common ETSE methods in terms of communication requirements. In particular, whenever the dynamics are obtained through learning methods, e.g., Gaussian process regression [12], or neural networks [13], these models can be huge, since they incorporate the whole training data set or rely on many parameters. Therefore, transmission of the learned function f might be infeasible.

Contributions: In summary, this paper makes the following main contributions:

- Proposing a local encoding of dynamical systems, based on a MDL-type objective, which uses basis expansions to achieve adaptive model complexities (e.g., the number of terms in Taylor expansion);
- Based on this, the development of the *Minimum Information Exchange in Distributed Systems* (MIEDS) algorithm, combining two ideas (i) local approximations of the function f ; and (ii) optimal choice for the accuracy of each local estimation;
- Application to ETSE for nonlinear stochastic systems and thus extending the MIEDS algorithm to stochastic problems;
- Proposing a unifying framework for common ETSE methods.

We focus on the information exchange between one sending and one receiving agent. Extending our results to multiple agents is possible, but requires extra work and is beyond the scope of the current paper.

Related work: Considering communication in networked control systems as a limited resource gave rise to various event-triggered communication architectures for estimation and control. In particular, event-triggered state estimation and control are well studied for linear dynamical systems; for recent overviews see, e.g., [14]–[17] and references therein.

In this paper, we mainly focus on remote state estimation, as in Fig. 1, for nonlinear dynamical systems. There are some results in ETSE for nonlinear systems [18], [19], however, they utilize a fixed global dynamical function f in order to predict future behavior on a remote side. Furthermore, there is little work that considers the transmission of dynamical models in addition to states. For linear systems, this problem has been addressed in [11] together with quantifying expected communication rates.

The MDL principle is a practical realization of Occam’s razor [1] and seeks for a model with a short description, which also yields compact and well behaved data. This

compression of data finds applications in various branches of statistics and machine learning such as statistical inference [20], unsupervised learning [21], Bayesian model selection [22], and causal inference [23]. The information content of dynamical systems is investigated in [24], [25].

The benefits of approximating dynamics models through local representations have long been recognized [26], [27]. While in these works, the order of the local models is usually fixed (e.g., linear), we propose herein to optimally choose the model complexity. Thus, we typically obtain models of different order for different state space regions.

II. PROBLEM FORMULATION

Consider a deterministic dynamical system

$$\dot{x}(t) = f(x(t)), \quad x(t_0) = x_0, \quad (1)$$

within a finite time horizon $t \in [t_0, t_0 + T]$. We assume f is a known nonlinear, Lipschitz continuous function, and the states $x(t) \in \Omega \subset \mathbb{R}^n$ can be measured. The system (1) can also represent a controlled system by incorporating the effect of some state feedback $u(t) = g(x(t))$ within f .

In this paper, we develop a tractable algorithm to encode the information of the dynamical system (1) efficiently. We assume that data from this system is available as a time-sampled signal $x(1), x(2), \dots$. Furthermore, we want to point out that it is not possible to transmit the full stream of continuous data and mention the slight abuse of notation in the indexing here.

Therefore, one possibility to represent the considered system is by storing all states at every time instance, i.e., $[x(1), x(2), \dots, x(T)]$. Alternatively, it is also possible to store the function f and an initial state $x(0)$. With the aid of the dynamical function, it is then possible to recover the trajectory of states and thus, data is compressed. However, it is in general not clear how to store a nonlinear function f in an optimal way. Therefore, we propose to approximate the function f through \hat{f} with the aid of a finite predefined basis ϕ_1, \dots, ϕ_k (to be made precise later), which yields the system

$$\dot{\hat{x}}(t) = \hat{f}(\hat{x}(t)) = \sum_{i=0}^k \alpha_i \phi_i(\hat{x}), \quad \hat{x}(t_0) = x_0, \quad (2)$$

where the α_i are weights. Since the dynamical function is used to recover a trajectory of states, approximating it by \hat{f} introduces an error to the corresponding state sequence. Choosing an optimal complexity k will be addressed in this paper. Furthermore, we propose to use local approximations $\hat{f}_1, \dots, \hat{f}_m$, with varying complexities. The choice of the optimal number of local models m will also be discussed later.

III. THE MDL PRINCIPLE FOR DYNAMICAL SYSTEMS

In this section, we introduce an MDL-type objective, which is the core of the proposed encoding scheme. In particular, it quantifies the trade-off between model complexity and accuracy. The resulting encoding scheme we denote as the MIEDS algorithm.

Minimum description length— The minimum description length (MDL) principle is a formalization of the well investigated trade-off between model complexity and accuracy in explaining data. First, we will explain the MDL principle in general for a given data set $D = \{(z_1, g(z_1)), \dots, (z_n, g(z_n))\}$ and a hypothesis space of models or mechanisms \mathcal{M} . The goal is finding the best model $\hat{f} \in \mathcal{M}$, which explains the data as accurately as possible, while still being simple. We quantify this trade-off with the objective function

$$\min_{\hat{f} \in \mathcal{M}} \left(L_{\text{comp}}(\hat{f}) + L_{\text{pred}}(D|\hat{f}) \right), \quad (3)$$

where the first term L_{comp} is a measure of complexity of the function \hat{f} and the second term L_{pred} is the prediction accuracy provided by this candidate function. As an example, assume a regression task, where we want to fit a polynomial function. In this case, L_{pred} measures the accuracy of the fit and L_{comp} is the order of the polynomial.

For systems of the type (1), the relevant data set is $D = \{(t_1, x_1), \dots, (t_n, x_n)\}$. The difference with the usual formalism of MDL is that instead of explicitly approximating the function of interest $x(t)$ by $\hat{x}(t)$, we approximate the right-hand side function f of the dynamical system (1) by \hat{f} . This gives us an implicit approximation of $x(t)$, which is produced by simulating a dynamical system from $x(0)$ via \hat{f} rather than f . This way, we efficiently compress information about the dynamical system (1).

The MDL-type objective functional— In order to find the appropriate complexity of the local models, we use the MDL principle and propose to optimize the following functional

$$\min_{k \in \{1, \dots, k_{\max}\}} \lambda k + \int_0^T \|x(t) - \hat{x}(t)\|_2 dt, \quad (4)$$

which is a concrete realization of (3). Here, $\hat{x}(t)$ is defined as in (2), and k is the number of terms we use for the approximation, which directly influences the accuracy of $\hat{x}(t)$. We only need to store or communicate the weights α in front of the basis functions ϕ , since these are assumed fixed a priori. The parameter $\lambda > 0$ can influence the relative weight of the model complexity. Together with the partitioning of the state space, this affects what kind of functions are preferred by the algorithm.

Intuition behind local models— What we eventually want to store, is the history of states of a dynamical system. For many practical systems, the states remain in a small subset of the entire state space, which makes the local approach to encoding powerful. Therefore, we care about a local representation of a function based on a finite set of basis functions where \hat{f} is the local approximation to f around a working point x^* . The set $\{\phi_i\}$ is chosen with cardinality k_{\max} to contain basis functions that are able to approximate f arbitrarily well as $k_{\max} \rightarrow \infty$. There are many well known types of basis functions and expansions that could be used such as Taylor series, Fourier series, or Legendre polynomials [28]. In this paper, we use Taylor expansions to showcase our points and thus assume f is smooth enough.

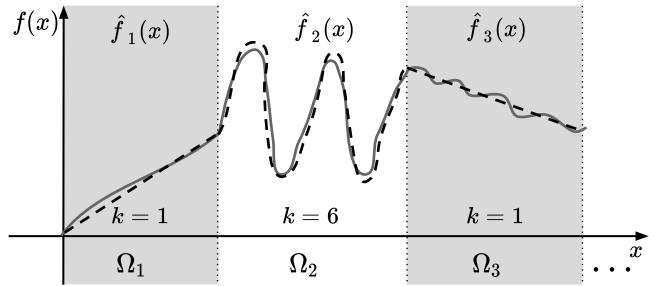


Fig. 2. Illustration of the proposed encoding for dynamical functions. The real dynamics (solid line) and local approximations (dashed line) are depicted. The state space is partitioned into disjoint sets Ω_i , in which the corresponding local approximation \hat{f}_i is valid. Based on a minimum description length type objective the local complexity order k of a Taylor expansion is determined. While the graphs are sketched to highlight key aspects of the approach, the same properties are observed in the numerical examples herein.

The information content of the full function f can be distributed to many local approximations. One advantage of local models is the ability to sample them fast. When facing demanding and computationally expensive control or filtering problems, this might be a key aspect to achieve online applicability. Furthermore, due to physical limitations or restrictions, the states $x(t)$ may stay within some bounded domain. Hence, the potential global input domain Ω reduces to the *effective* input space, which we eventually care about.

Figure 2 sketches the main idea behind the local approximations. In the Ω_1 part of the state space, the function is almost linear and hence, a linear approximation is used ($k=1$). In Ω_2 , however, there are significant nonlinearities and therefore, a polynomial with degree $k=6$ is chosen. A different effect can be seen in Ω_3 , where a linear model is preferred, because here, a marginal improvement in the states does not justify the cost caused by a more complex model. This trade-off, which is quantified in (4), is essential in order to determine automatically the required model complexity.

Partitioning and local models— We use an equidistant partitioning of the state space, and the number of necessary partitions is determined through the proposed algorithm. Furthermore, we require that the state space is bounded. However, both assumptions could be relaxed by checking the distance between $x(t)$ and $\hat{x}(t)$ and refining the partitions on necessity. Applying other more sophisticated methods is also possible here.

Here, we define the maximal number of partitions m_{\max} as a hyperparameter, which should be chosen large enough. We propose to partition the time domain and thus $T_{\text{local}} = \frac{T}{m}$. Clearly, the local time horizon gets shorter as the number of partitions grows. We iterate over all possible choices of m by locally solving for the optimal value of the cost functional (4) and summing over all local costs, which yields

$$\arg \min_{m \in \{1, \dots, m_{\max}\}} L_{\text{total}}(m) = \sum_{i=1}^m L_i(k_i^*). \quad (5)$$

The local optimal complexity k_i^* is obtained by optimizing each individual local cost over the time horizon $T_{\text{local}} =$

$t_i^{\text{stop}} - t_i^{\text{start}}$. Therefore, we obtain

$$L_i(k_i) = \lambda k_i + \int_{t_i^{\text{start}}}^{t_i^{\text{stop}}} \|x(t) - \hat{x}(t)\|_2 dt. \quad (6)$$

Remark 1: The local times also induce a partitioning in space, which we denote with Ω_i . We obtain the Ω_i by evaluating the state trajectory at the desired switching point in time. It is also possible to define the partitioning in the state space and afterwards compute the corresponding times. To summarize, the complete encoding scheme consists of fixing a partitioning of the state space and finding optimal local approximations with respect to the introduced MDL-type objective functional (6). Afterwards, we change the partitioning, e.g., increase m , and optimize again the local cost. Finally, we obtain the optimal m^* and the corresponding models $\hat{f}_1, \dots, \hat{f}_m$ with varying complexities k_1^*, \dots, k_m^* . We call this the Minimum Information Exchange in Distributed Systems (MIEDS) algorithms due to its motivation from information exchange between distributed agents (Fig. 1). In the next section, we demonstrate nontrivial behavior of m and k in numerical examples.

IV. EXPERIMENTS: ENCODING

We provide numerical experiments to demonstrate different aspects of the MIEDS algorithm in encoding a deterministic dynamical system.

Pendulum dynamics— To start with, we consider the dynamics of a pendulum with two state equations $\dot{x}_1 = x_2$ and $\dot{x}_2 = -x_2 - 9.81\sin(x_1)$, where x_1 is the angular position and x_2 the angular velocity. The system starts from the initial state $[x_1(0), x_2(0)] = [\frac{\pi}{4}, 0]$. Assume polynomials with degrees up to 3 are allowed as the set of basis functions to approximate each local region. Given this set, the approximation is done by Taylor expansions. The optimal degree k_i of the Taylor expansion corresponding to each local region can be seen in Fig. 3. We choose the parameters $\lambda = 2$, $T = 2$, and $k_{\max} = 3$. The sampling time step ΔT is set to 0.01.

Quadrotor dynamics— One typical example where dynamical system information is transmitted between agents, are Unmanned Aerial Vehicles (UAVs), whose states are constantly measured on-board, but only occasionally transmitted to the ground base. Continuous transmission of data is expensive in terms of battery power and bandwidth. To show the performance of MIEDS on more complex dynamics, we present a dynamical system of a quad-rotor UAV [29].

In the following state space equation, $\phi \in [-\pi, \pi]$ is the roll (rotation around x axis) and $\theta \in (-\frac{\pi}{2}, \frac{\pi}{2})$ is the pitch (rotation around y axis). The value of angles can violate the bounds but its interpretation is circular. The vector $[u, v, w, p, q, r]$ contains the linear and angular velocities in the body frame. The vector $[f_{wx}, f_{wy}, f_{wz}, f_t] = [1, 1, 1, 0]$ contains the wind forces and time-varying disturbance, $[\tau_{wx}, \tau_{wy}, \tau_{wz}] = [1, 1, 1]$ wind torques, and $[\tau_x, \tau_y, \tau_z] = [1, 1, 1]$ the control torques generated by the differences in the rotor speeds. We assume a mass $m = 1$ and set the

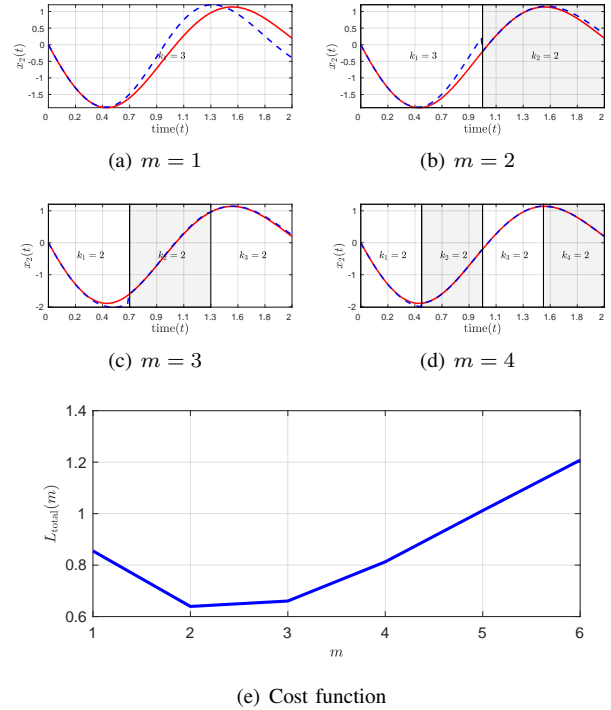


Fig. 3. Pendulum dynamics. The top four figures show the system trajectory of the state $x_2(t)$ in red and different approximations resulting from our method in blue. Depending on the used partitions m and Taylor terms k we can observe different behavior. The depicted cost function in Panel (e) shows a nontrivial optimal value $m^* = 2$.

gravity constant to $g = 9.81$. This yields the system

$$\begin{cases} \dot{\phi} = p + r(\cos(\phi) \tan(\theta)) + q(\sin(\phi) \tan(\theta)) \\ \dot{\theta} = q(\cos(\theta)) - r(\sin(\phi)) \\ \dot{p} = \frac{I_y - I_z}{I_x} r q + \frac{\tau_x + \tau_{wx}}{I_x} \\ \dot{q} = \frac{I_z - I_x}{I_y} p r + \frac{\tau_y + \tau_{wy}}{I_y} \\ \dot{r} = \frac{I_x - I_y}{I_z} p q + \frac{\tau_z + \tau_{wz}}{I_z} \\ \dot{u} = r v - q w - g \sin(\theta) + \frac{f_{wx}}{m} \\ \dot{v} = p w - r u + g \sin(\phi) \cos(\theta) + \frac{f_{wy}}{m} \\ \dot{w} = q u - p v + g \cos(\theta) \cos(\phi) + \frac{f_{wz} - f_t}{m} \end{cases} \quad (7)$$

with an initial state $[-2, -3, 1, 3, 1, 4, 2, 1]$.

In Fig. 4, we can see the effects of different model complexities on the state prediction accuracy. We take polynomials up to degree 5 as allowed basis functions and the maximum number of partitions is upper bounded by 5. Hence $k_{\max} = 5$ and $m_{\max} = 5$. The parameters λ, T , and ΔT are chosen as before.

V. APPLICATION TO ETSE

In this section, we demonstrate the effectiveness of our encoding scheme in ETSE scenarios, as motivated in the introduction. Because ETSE approaches typically consider stochastic systems, we first generalize MIEDS to the stochastic setting (called stochastic MIEDS (sMIEDS)).

A. Extension to Stochastic Systems

We consider stochastic dynamical systems of the type

$$\dot{x}(t) = f(x(t)) + \epsilon(t), \quad x(t_0) = x_0, \quad (8)$$

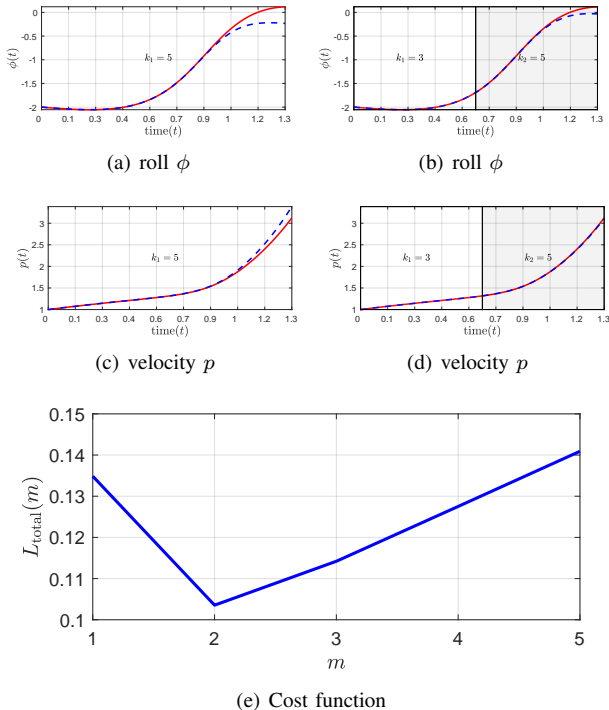


Fig. 4. Quadrotor dynamics. Top four figures show the effect of suboptimal ($m = 1$, left) and optimal number of partitions ($m = 2$, right) on two different states of the system. Having a single partition leads to complex models. Partitioning the state space into two local regions gives a fair trade-off between model complexity and data fit. The diagram (e) shows again a nontrivial optimum.

where ϵ is white noise and f is a nonlinear function, which satisfies the conditions for the existence and uniqueness of solutions to stochastic differential equations (see e.g., [30]).

We adjust Equation 6 by introducing a new variable $\bar{x}(t)$, which solves the deterministic problem without noise, which yields the local cost

$$L_i(k_i) = \lambda k_i + \int_{t_i^{\text{start}}}^{t_i^{\text{stop}}} \|\bar{x}(t) - \hat{x}(t)\|_2 dt. \quad (9)$$

The partitioning should depend on the deterministic effects that drive the system and not on noise effects. Furthermore, the stochastic realizations vary between roll outs and we care more about the average behaviour of the system.

B. Event-triggered state estimation

We introduce a typical ETSE framework, in order to apply our encoding scheme to it afterwards. Consider the problem in Fig. 1, where we have two agents, one sending v_{send} and one receiving v_{receive} . We assume the behavior of agent v_{send} is described by (8). At the same time, the remote agent v_{receive} needs information about those states for tasks such as monitoring or control. We assume it is possible, but costly to communicate states through a network. Instead of transmitting data periodically, the receiving agent v_{receive} utilizes a model to obtain approximations $\hat{x}(t)$ of the true states $x(t)$. In order to ensure sufficient accuracy, we deploy the trigger $\gamma_{\text{noise}} \in \{0, 1\}$,

$$\gamma_{\text{noise}} = 1 \quad \Leftrightarrow \quad \|\hat{x}(t^*) - x(t^*)\| \geq \delta_{\text{noise}}. \quad (10)$$

If the trigger fires ($\gamma_{\text{noise}} = 1$), v_{send} transmits the current state, and v_{receive} resets the prediction, $\hat{x}(t^*) \leftarrow x(t^*)$ and t^* denotes the corresponding point in time. The sending agent v_{send} implements (10) and thus also runs the same computations as v_{receive} to obtain $\hat{x}(t)$. Due to the stochastic nature of the system, predictions can go arbitrarily bad, but (10) enforces a bound on the error. This is a common scheme in ETSE (cf. *Related work* in Sec. I).

Beginning from the most recently seen state $x(t^*)$, the receiving agent v_{receive} computes

$$\hat{x}(t) = \mathbb{E}[x(t)|x(t^*)], \quad t > t^*, \quad (11)$$

where $\hat{x}(t^*) = x(t^*)$. Most ETSE schemes utilize the dynamics function f in order to obtain $\hat{x}(t)$. For linear systems with Gaussian noise, the predictions (11) can be obtained analytically. However, for nonlinear systems, the computation is usually intractable and approximations are required. Here, we implement the approximation scheme

$$\dot{\hat{x}}(t) = \hat{f}(\hat{x}(t)), \quad \hat{x}(t_0) = x(t^*), \quad (12)$$

which is not optimal, but straightforward and directly illustrates the importance of good models \hat{f} . As an alternative to (12), particle filter type methods can be used, to obtain an unbiased estimate. However, these methods suffer immensely from inapt model representations, since they require to sample from the model repeatedly. Efficient local models help here a lot, in particular when we consider time sensitive online applications.

The MIEDS algorithm yields a natural way to consider local models as transmittable data as well and yields, due to the MDL approach, a principled way of choosing optimal model complexities. We propose to transmit new models in an event-triggered fashion and thus reduce overall communication.

C. Design of dual triggering scheme

Next, we describe how to decouple stochastic effects from deterministic model errors, due to local approximations. To integrate the MIEDS algorithm into ETSE, we propose to consider two triggers – one to capture local random fluctuations γ_{noise} (10), and one to cope with the change in the local dynamics γ_{dynamics} . Whenever the trigger γ_{noise} fires, we communicate solely the current state. However, the trigger γ_{dynamics} is responsible to detect model inaccuracies, because of entering a new region Ω_i and communicate new local models when required.

The trigger γ_{dynamics} is derived from applying the MIEDS algorithm to the deterministic system $\dot{\hat{x}}(t) = f(\hat{x}(t))$. This way, we obtain local models \hat{f}^i , each with a local domain Ω_i . Assume \bar{t} corresponds to T_{local} and is the point in time when switching should happen. By evaluating $x(\bar{t})$, we obtain a switching point and illustrate this in Sec. VI for a one dimensional example.

Instead of switching the models after a predefined time, we emphasize the event triggered nature of our approach. Due to the noise, the system might stay arbitrary long within a local domain and the state dependent model switching is

more robust to noise. The trigger $\gamma_{\text{dynamics}} \in \{0, 1\}$ fires whenever

$$x(t) \notin \Omega_i \Leftrightarrow \gamma_{\text{dynamics}} = 1. \quad (13)$$

Hence, the local models are actually switched when necessary, based on the current state and not the current time. Depending on the application, it may be necessary to trigger on $\hat{x}(t)$ as well, or shrink the domain Ω_i , since we know that $x(t)$ and $\hat{x}(t)$ are at most δ_{noise} apart from each other. By running the two trigger γ_{noise} and γ_{dynamics} in parallel, we obtain the stochastic MIEDS (sMIEDS) algorithm, which utilizes the advantages of local models in a stochastic setting.

D. Interpretation of existing ETSE schemes

Standard event-triggered schemes can be understood in the sense of the MDL principle as well. Most protocols trigger communication whenever $\|x(t) - \hat{x}(t)\| \geq \delta$, however, the computation of $\hat{x}(t)$ may differ for different strategies. The send-on-delta (SoD) concept uses the latest communicated state, which means $\hat{x}(t^*) = x(t^*)$. This coincides with predictions with the identity and hence, corresponds to the hypothesis space $\mathcal{M} = \{\text{Id}\}$. For linear dynamics $f(x) = Ax$, we obtain $\mathcal{M} = \{A : A \in \mathbb{R}^{n \times n}\}$. Usually, a fixed model is utilized to run the predictions, however, there are also approaches, where the dynamical model is actually chosen from that set through linear regression, as in [11].

With the above interpretation, the MIEDS formalism allows us to extend ETSE to the nonlinear setting. For nonlinear dynamical systems, we assume the hypothesis space is given by a basis set of a suitable function space, e.g., a power series or Fourier series. Hence, we obtain $\mathcal{M} = \{\hat{f} : \hat{f}(x) = \sum_{i=0}^k \alpha_i \phi(x)\}$, where α_i are coefficients and ϕ_i basis functions. By communicating models in addition to states, we impose additional costs in the ETSE setting.

VI. EXPERIMENTS: REDUCED COMMUNICATION

In this section, we present numerical experiments to demonstrate reduced communication behavior in an ETSE setting. We consider the stochastic system $\dot{x}(t) = -\tanh(x(t)) + \frac{1}{10}\epsilon(t)$, where $\epsilon(t)$ is standard white noise. We assume $x(0) = 6$ and use the Euler-Maruyama method [30] to sample from the system. We consider a fixed time horizon of $T = 10$ s and check according to the MIEDS algorithm different numbers of partitions of the state space. For the choice of $\lambda = 0.01$, we obtain $m^* = 2$ and hence, two local domains. We determine the switching point by evaluating the solution to $\dot{\hat{x}}(t) = -\tanh(\bar{x}(t))$ in $t = 5$ s. The local models are obtained with the aid of Taylor expansions and the MIEDS algorithm yields a linear function ($k_1 = 1$) within Ω_1 and a cubic polynomial ($k_2 = 3$) within Ω_2 .

In this experiment, we compare performance based on the number of state transmissions due to the trigger γ_{noise} in (10). In particular, we compare different choices of \hat{f} in Equation (12):

- Analytical predictions $\hat{x}^{\text{Analytical}}(t)$, based on the analytical model $f(x) = -\tanh(x)$;

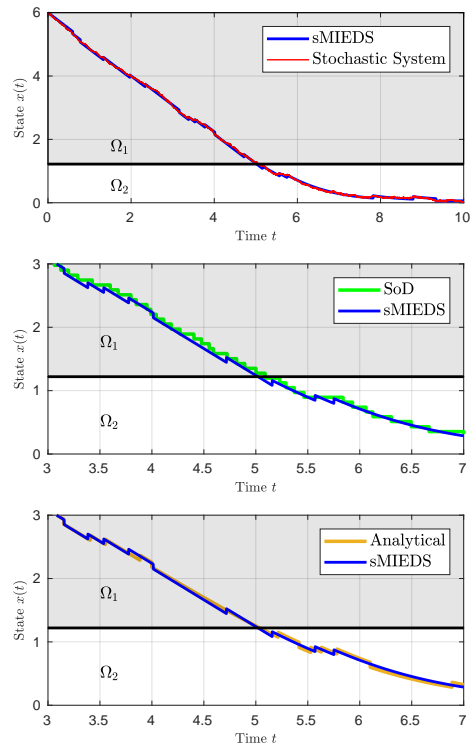


Fig. 5. Stochastic example, for which we demonstrate the benefits of our approach. The first plot shows the stochastic trajectory of the system in red and our method in blue. Every jump corresponds to communication due to noise effects. The local model is updated when the horizontal line is crossed. In the second plot we see an improvement in communication savings compared to the send-on-delta protocol, which is highlighted by zooming in. Furthermore, we show in the third plot similar performance with respect to the analytical model, with the addition benefit of significantly reducing the model complexity.

- sMIEDS predictions $\hat{x}^{\text{sMIEDS}}(t)$, based on the local approximations \hat{f}^i ;
- Send-on-delta predictions \hat{x}^{SoD} , which use a zero-order hold [10] and therefore, $\hat{f} \equiv 0$.

Hence, for every fixed trajectory of the stochastic system $x(t)$, we compute the three predictions $\hat{x}^{\text{Analytical}}(t)$, $\hat{x}^{\text{sMIEDS}}(t)$, and $\hat{x}^{\text{SoD}}(t)$ and enforce the trigger (10) with $\delta_{\text{noise}} = 0.075$. In Fig. 5, we illustrate one trajectory of the system (in red). In the first subfigure, we see the predictions $\hat{x}^{\text{sMIEDS}}(t)$ (in blue). Every jump corresponds to transmitting the current state. The other two plots show a zoom in, to yield a direct comparison to the two benchmarks (send-on-delta in green and analytical model in orange). Due to the stochastic nature of the system, we run 100 simulations with the same parameters, initial conditions, and time horizon. Then, we average the number of communication instances and obtain

$$\gamma_{\text{noise}}^{\text{SoD}} \approx 78, \quad \gamma_{\text{noise}}^{\text{Analytical}} \approx 15, \quad \gamma_{\text{noise}}^{\text{sMIEDS}} \approx 16.$$

Clearly, the SoD-architecture requires significantly more communication and demonstrates that for this example, our algorithm saves communication by a factor of 5. On the other hand, the performance of the analytical model is very comparable to our algorithm. Hence, we show that we are close to optimal in terms of the number of communication

instances, while decreasing model complexity significantly. The cost of transmitting a model is not considered here and would result in $4 = k_1 + k_2$ additional weights that need to be sent. In the beginning, for the initial model, and when the black line separating Ω_1 from Ω_2 is crossed. For the SoD protocol this would not be necessary.

VII. DISCUSSION

Using ideas from the MDL principle, the MIEDS algorithm is developed herein and proposed as a novel concept to optimally encode dynamical systems. We empirically show that the optimal encoding happens somewhere between two extremes of a spectrum: either storing the complete state history, or an initial state and the whole dynamics function. The MIEDS algorithm is applied to multi-dimensional deterministic systems to illustrate the core ideas and extended to cope with stochastic systems (sMIEDS). While the method introduces some computational overhead to compute the local approximations, we demonstrate improvements in terms of more efficient representations. In particular, for event-triggered state estimation, we show the net benefit in terms of reduced communication. This paper is a proof of concept that information from dynamical systems can be efficiently encoded through local dynamics approximations. However, there are still many challenges that need to be addressed in future work.

Generalizing the ideas developed herein to high-dimensional stochastic and multi-agent problems is an possible next step. While the main ingredients for this are developed in this paper, demonstrating the benefits (e.g., communication savings) for applications at the scale of real-world problems is still to be shown. Theoretically quantifying the introduced error to state trajectories by approximating the dynamical function is also a topic of ongoing research. While we assume the dynamics function f to be known, it could likewise be learned from data. In particular, for nonparametric techniques, where the learned function is typically represented by a large data set, the proposed techniques are relevant.

SUPPLEMENTARY MATERIAL

Matlab implementations are available at: <https://github.com/amehrjou/CDC2018>

REFERENCES

- [1] A. Barron, J. Rissanen, and B. Yu, "The minimum description length principle in coding and modeling," *IEEE Transactions on Information Theory*, vol. 44, no. 6, pp. 2743–2760, 1998.
- [2] V. Vapnik, *The nature of statistical learning theory*. Springer science & business media, 2013.
- [3] E. L. Ionides, C. Bretó, and A. A. King, "Inference for nonlinear dynamical systems," *Proceedings of the National Academy of Sciences*, vol. 103, no. 49, pp. 18438–18443, 2006.
- [4] W. L. Hamilton, M. M. Fard, and J. Pineau, "Modelling sparse dynamical systems with compressed predictive state representations," in *International Conference on Machine Learning*, 2013, pp. 178–186.
- [5] J. K. Yook, D. M. Tilbury, and N. R. Soparkar, "Trading computation for bandwidth: reducing communication in distributed control systems using state estimators," *IEEE Transactions on Control Systems Technology*, vol. 10, no. 4, pp. 503–518, Jul. 2002.
- [6] S. Trimpe and R. D'Andrea, "An experimental demonstration of a distributed and event-based state estimation algorithm," in *18th IFAC World Congress*, 2011, pp. 8811–8818.
- [7] J. Sijts and M. Lazar, "Event based state estimation with time synchronous updates," *IEEE Transactions on Automatic Control*, vol. 57, no. 10, pp. 2650–2655, 2012.
- [8] S. Trimpe and R. D'Andrea, "Event-based state estimation with variance-based triggering," *IEEE Transactions on Automatic Control*, vol. 59, no. 12, pp. 3266–3281, 2014.
- [9] J. Wu, Q.-S. Jia, K. H. Johansson, and L. Shi, "Event-based sensor data scheduling: Trade-off between communication rate and estimation quality," *IEEE Transactions on Automatic Control*, vol. 58, no. 4, pp. 1041–1046, 2013.
- [10] M. Miskowicz, "Send-on-delta concept: an event-based data reporting strategy," *Sensors*, vol. 6, no. 1, pp. 49–63, 2006.
- [11] F. Solowjow, D. Baumann, J. Garcke, and S. Trimpe, "Event-triggered learning for resource-efficient networked control," in *Proceedings of the 2018 American Control Conference (ACC)*, 2018.
- [12] A. Doerr, C. Daniel, D. Nguyen-Tuong, A. Marco, S. Schaal, M. Toussaint, and S. Trimpe, "Optimizing long-term predictions for model-based policy search," in *Conference on Robot Learning*, 2017.
- [13] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Transactions on neural networks*, vol. 1, no. 1, pp. 4–27, 1990.
- [14] D. Shi, L. Shi, and T. Chen, *Event-Based State Estimation: A Stochastic Perspective*. Springer, 2015.
- [15] S. Trimpe and M. C. Campi, "On the choice of the event trigger in event-based estimation," in *Int. Conf. on Event-based Control, Communication, and Signal Processing*, 2015, pp. 1–8.
- [16] W. P. M. H. Heemels, K. H. Johansson, and P. Tabuada, "An introduction to event-triggered and self-triggered control," in *Proc. of the IEEE 51st Conference on Decision and Control*, 2012, pp. 3270–3285.
- [17] M. Miskowicz, *Event-Based Control and Signal Processing*. CRC Press, 2016.
- [18] S. Trimpe and J. Buchli, "Event-based estimation and control for remote robot operation with reduced communication," in *IEEE Int. Conf. on Robotics and Automation*, 2015, pp. 5018–5025.
- [19] M. Martínez-Rey, F. Espinosa, A. Gardel, and C. Santos, "On-board event-based state estimation for trajectory approaching and tracking of a vehicle," *Sensors*, vol. 15, no. 6, pp. 14569–14590, 2015.
- [20] V. Balasubramanian, "Statistical inference, occam's razor, and statistical mechanics on the space of probability distributions," *Neural computation*, vol. 9, no. 2, pp. 349–368, 1997.
- [21] R. S. Zemel, *A minimum description length framework for unsupervised learning*. University of Toronto, 1994.
- [22] A. Mehrjou, R. Hosseini, and B. N. Araabi, "Improved bayesian information criterion for mixture model selection," *Pattern Recognition Letters*, vol. 69, pp. 22–27, 2016.
- [23] D. Janzing and B. Schölkopf, "Causal inference using the algorithmic markov condition," *IEEE Transactions on Information Theory*, vol. 56, no. 10, pp. 5168–5194, 2010.
- [24] D. Balduzzi and G. Tononi, "Integrated information in discrete dynamical systems: motivation and theoretical framework," *PLoS computational biology*, vol. 4, no. 6, 2008.
- [25] R. Kleeman, "Information theory and dynamical system predictability," *Entropy*, vol. 13, no. 3, pp. 612–649, 2011.
- [26] C. G. Atkeson, A. W. Moore, and S. Schaal, "Locally weighted learning for control," in *Lazy learning*. Springer, 1997, pp. 75–113.
- [27] O. Nelles and R. Isermann, "Basis function networks for interpolation of local linear models," in *Decision and Control, 1996., Proceedings of the 35th IEEE Conference on*, vol. 1. IEEE, 1996, pp. 470–475.
- [28] H. W. Alt, "Linear functional analysis," *An application oriented introduction*, 1992.
- [29] F. Sabatino, "Quadrotor control: modeling, nonlinear control design, and simulation," 2015.
- [30] Z. Schuss, *Theory and applications of stochastic processes: an analytical approach*. Springer Science & Business Media, 2009.