PAPER

Learning Dynamic Systems Using Gaussian Process Regression with Analytic Ordinary Differential Equations as Prior Information

Shengbing TANG^{†a)}, *Nonmember*, Kenji FUJIMOTO^{†b)}, *Member*, and Ichiro MARUTA^{†c)}, *Nonmember*

SUMMARY Recently the data-driven learning of dynamic systems has become a promising approach because no physical knowledge is needed. Pure machine learning approaches such as Gaussian process regression (GPR) learns a dynamic model from data, with all physical knowledge about the system discarded. This goes from one extreme, namely methods based on optimizing parametric physical models derived from physical laws, to the other. GPR has high flexibility and is able to model any dynamics as long as they are locally smooth, but can not generalize well to unexplored areas with little or no training data. The analytic physical model derived under assumptions is an abstract approximation of the true system, but has global generalization ability. Hence the optimal learning strategy is to combine GPR with the analytic physical model. This paper proposes a method to learn dynamic systems using GPR with analytic ordinary differential equations (ODEs) as prior information. The one-time-step integration of analytic ODEs is used as the mean function of the Gaussian process prior. The total parameters to be trained include physical parameters of analytic ODEs and parameters of GPR. A novel method is proposed to simultaneously learn all parameters, which is realized by the fully Bayesian GPR and more promising to learn an optimal model. The standard Gaussian process regression, the ODE method and the existing method in the literature are chosen as baselines to verify the benefit of the proposed method. The predictive performance is evaluated by both one-time-step prediction and long-term prediction. By simulation of the cart-pole system, it is demonstrated that the proposed method has better predictive performances.

key words: Gaussian process regression (GPR), prior information, analytic ordinary differential equations, dynamic systems

1. Introduction and Related Work

Obtaining an accurate model of the dynamic system [1]–[3] is essential for many applications, such as controller design and model-based reinforcement learning. Gaussian process regression (GPR) is a promising method for learning dynamic systems using training data because of its high flexibility [4], [5]. For example, GPR is used to model the state transition dynamics for model-based reinforcement learning [6], [7]. The idea of learning dynamic systems with standard GPR is as follows: at first assume that the prior distribution of the unknown system function is a zero-mean Gaussian process; then the posterior distribution is also a Gaussian process conditioned on training data. This is a pure machine learning approach which totally ignores the prior physical knowledge about the dynamic system. The major

advantage of GPR is the ability to model any dynamics as long as they are locally smooth. However GPR has poor ability of extrapolation, namely it can not generalize well to regions where there is little or no training data. Thus to obtain an accurate dynamic model, the training data should cover the entire or at least important areas of the state-action space, which becomes exponentially difficult as the dimensions of state and action increase.

In order to alleviate above problems, semi-parametric models which combine GPR with analytic models of dynamic systems have been studied. Firstly the literature in the field of learning forward dynamics is introduced. The work in [8] models forward dynamics of a blimp by combining the ODE model with the standard GPR. At first the training data is used to optimize physical parameters of the ODE model; then a zero-mean GPR is used to model the residual between the prediction of the optimized ODE model and training data. The combined model outperforms the standard GPR which demonstrates the benefit of making use of physical information of the dynamic system. However, physical parameters of the ODE model and hyperparameters of GPR are learned separately, which may yield a suboptimal model. In [9], [10], instead of discrete-time, continuous-time state transition dynamics are learned. The GPR is used to learn the mapping from positional state and action to acceleration. In order to utilize the prior knowledge of the dynamic system, the equation of motion is derived from the Lagrange approach, and then the acceleration function is formalized as a linear-in-parameter model, which is used as the mean function of the Gaussian process prior. The equations described in [4] (chapter 2.7) are used to marginalize out the parameters of the mean function. The strong assumption of the linear-in-parameter mean function greatly limits its application. The work in [11] uses data from arbitrary simulators of the real-world dynamic system as prior information to learn forward dynamics, which can decrease the number of samples needed by the learning algorithm to achieve the desired accuracy. At first the simulator of the real-world physical system is built, with parameters roughly measured or estimated. Data generated from the simulator is used to learn a Gaussian process model of the simulated transition dynamics. The mean of simulated transition dynamics is then used as the mean function of the Gaussian process prior of real-world transition dynamics. The parameters of the simulator need to be obtained without any training data, which may be difficult and impractical in some situations. Furthermore, the performance of the algorithm depends heavily on the ac-

Manuscript received September 24, 2020.

Manuscript revised March 22, 2021.

Manuscript publicized June 1, 2021.

[†]The authors are with the Department of Aeronautics and Astronautics, Kyoto University, Kyoto-shi, 615–8540 Japan.

a) E-mail: tang.shengbing.77w@st.kyoto-u.ac.jp

b) E-mail: k.fujimoto@ieec.org

c) E-mail: maruta@kuaero.kyoto-u.ac.jp

DOI: 10.1587/transinf.2020EDP7186

curacy of the simulator. Data generated from poor simulators may be harmful to the learning process.

Secondly the literature in the field of learning inverse dynamics is introduced. The works in [12]–[15] learn inverse dynamics of robots by combining physical models with the GPR. For robots the inverse dynamics can be abstractly described by the rigid body dynamic (RBD) model which is linear-in-parameter. To incorporate the prior information of the dynamic system into the GPR learning framework, the RBD model is used as the mean function of the Gaussian process prior. In [12], [13], the authors discuss the method of incorporating the RBD model into the mean function or the kernel function of Gaussian process respectively.

This paper proposes a method to learn dynamic systems using GPR with analytic ODEs as prior information. The core idea is to use the one-time-step integration of analytic ODEs as the mean function of the Gaussian process prior. By this way, the high flexibility of GPR and the global generalization ability of analytic ODEs is combined. As a result, the extrapolation ability of GPR will be improved. The total parameters to be trained include physical parameters of analytic ODEs, parameters of the kernel function and the noise variance. This paper proposes a novel method to simultaneously learn all parameters which is more promising to learn an optimal model. The fully Bayesian GPR is adopted, where all parameters are regarded as random variables and assigned prior distributions. The posterior distribution of all parameters is approximated by Markov Chain Monte Carlo. The standard Gaussian process regression, the ODE method and the existing method in the literature are chosen as baselines to verify the advantage of the proposed method. The contributions of the paper are summarized as follows:

- This paper proposes a more general framework to combine GPR with analytic ODEs to learn dynamic systems.
- A novel method is proposed to simultaneously learn all parameters, which is realized by the fully Bayesian GPR.

2. Problem Formulation

Learning the dynamic system is equivalent to estimating the unknown function $f(x)$ given a training set $\mathcal{D}$ of N samples, $\mathcal{D} = \{(x_i, y_i) | i = 1, 2, \dots, N\}$:

$$y = f(x) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2) \quad (1)$$

where the input x is the state-action pair; the output y is the consecutive state. ϵ is the zero-mean Gaussian noise. With training inputs collected as $X = \{x_1, x_2, \dots, x_N\}$, training outputs collected as $Y = \{y_1, y_2, \dots, y_N\}$, the training set $\mathcal{D}$ can be written as $\mathcal{D} = (X, Y)$. The standard GPR [4] for estimating the unknown function $f(x)$ is as follows. At first, assume that the prior distribution of the function $f(x)$ is zero-mean Gaussian process:

$$f(x) \sim \mathcal{GP}(0, k(x, x')). \quad (2)$$

Then the posterior distribution of $f(x)$ is still Gaussian process. The predictive distribution of noisy outputs Y_* at testing inputs X_* is a Gaussian distribution:

$$\begin{aligned} Y_* | \{X, Y, X_*\} &\sim \mathcal{N}(\mu_*, \Sigma_*), \quad \text{where}^\dagger \\ \mu_* &= K(X_*, X) [K(X, X) + \sigma^2 \mathbf{I}]^{-1} Y \\ \Sigma_* &= K(X_*, X_*) + \sigma^2 \mathbf{I} - K(X_*, X) [K(X, X) + \sigma^2 \mathbf{I}]^{-1} K(X, X_*). \end{aligned} \quad (3)$$

Here $K(X, X)$ is the kernel matrix with elements $K_{ij} = k(x_i, x_j)$.

The above learning process totally discards physical knowledge of the system which seems very unnatural because for most dynamic systems at least analytic ordinary differential equations can be derived from physical laws without any training data. Furthermore, the GPR is essentially a Bayesian estimation method, for which all available useful prior information should be extracted and incorporated into the prior distribution, especially in the case of small training sample size.

Suppose that analytic ODEs derived from physical laws are denoted as:

$$\dot{z} = s(z, u; w) \quad (4)$$

where w is the unknown physical parameter; z is the state; u is the action. Hence only the structure of analytic ODEs is known. Since it is difficult to model complex frictions and hysteresis, the analytic ODEs may deviate from the real dynamic system. But they still contain useful information.

3. Gaussian Process Regression with Analytic ODEs as Prior Information

This paper focus on discrete time dynamic systems. To make use of physical knowledge of the dynamic system, the one-time-step integration of analytic ODEs is used as the mean function of the Gaussian process prior. The one-time-step integration of analytic ODEs, denoted as the function h , behaves like a mapping from the current state-action pair (z_t, u_t) to the consecutive state z_{t+1} :

$$z_{t+1} = h(z_t, u_t; w). \quad (5)$$

Thus the function h is used as the mean function of the Gaussian process prior. Then the prior distribution of the unknown system function $f(x)$ is Gaussian process with the mean function $m(x; w) = h(x; w)$:

$$f(x) \sim \mathcal{GP}(m(x; w), k(x, x')). \quad (6)$$

The mean function $m(x; w)$ is parameterized by the physical parameter w . Denote the parameter of the kernel function as k_θ . Total parameters to be trained include $\{w, k_\theta, \sigma^2\}$. This paper proposes a novel method to simultaneously learn all

[†] $Y_* | \{X, Y, X_*\}$ refers to $p(Y_* | X, Y, X_*)$.

parameters $\{w, k_\theta, \sigma^2\}$.

3.1 Preliminary Method

The method in [8] consists of two steps: at first, the physical parameter w is learned by fitting analytic ODEs to training samples; then the kernel parameter k_θ and the noise variance σ^2 are learned by training a GPR with the fixed mean function. Obviously this method has the risk of learning a suboptimal model because the physical parameter w and the kernel parameter k_θ , the noise variance σ^2 are optimized separately.

The physical parameter w is optimized by minimizing the sum-of-square error:

$$\bar{w} = \underset{w}{\operatorname{argmin}} \sum_{i=1}^N \|h(x_i; w) - y_i\|^2. \quad (7)$$

The function value $h(x_i; w)$ is the one-time-step integration of analytic ODEs with the initial state and action given by the training input x_i . With the optimized physical parameter $\bar{w}$, the Gaussian process prior has the fixed mean function $m(x; \bar{w})$:

$$f(x) \sim \mathcal{GP}(m(x; \bar{w}), k(x, x')). \quad (8)$$

According to the approach in [4], training a GPR with a fixed mean function is equivalent to training a GPR with a zero mean function:

$$f(x) - m(x; \bar{w}) \sim \mathcal{GP}(0, k(x, x')). \quad (9)$$

The corresponding training set becomes $(x_i, y_i - m(x_i; \bar{w}))_{i=1}^N = (X, Y - m(X; \bar{w}))$. The remaining work is exactly the same as the standard GPR, with the kernel parameter k_θ and the noise variance σ^2 learned by evidence maximization. Finally the predictive distribution of noisy outputs Y_* at testing inputs X_* is:

$Y_* | \{X, Y, X_*\} \sim \mathcal{N}(\mu_*, \Sigma_*)$, where

$$\begin{aligned} \mu_* &= m(X_*; \bar{w}) + K(X_*, X) [K(X, X) + \sigma^2 \mathbf{I}]^{-1} (Y - m(X; \bar{w})) \\ \Sigma_* &= K(X_*, X_*) + \sigma^2 \mathbf{I} - K(X_*, X) [K(X, X) + \sigma^2 \mathbf{I}]^{-1} K(X, X_*). \end{aligned} \quad (10)$$

3.2 Proposed Method

This section proposes a novel method to train all parameters $\{w, k_\theta, \sigma^2\}$ simultaneously, which is more promising to learn an optimal model. The fully Bayesian treatment of GPR is adopted, where all hyperparameters are regarded as random variables and assigned prior distributions. The posterior distribution of all hyperparameters is approximated by Markov Chain Monte Carlo. The proposed method is a gradient-free optimization method, which is able to use the more general model as the mean function, such as complicated simulators with tunable parameters.

For the fully Bayesian GPR model, parameters are values of the unknown function $f(x)$ at training inputs $\{x_1, x_2, \dots, x_N\}$, which are collected as $\mathbf{f} =$

$(f(x_1), f(x_2), \dots, f(x_N))^T$. The hierarchical model is:

prior over hyperparameters: $w, k_\theta, \sigma^2 \sim p(w, k_\theta, \sigma^2)$

prior over parameters: $\mathbf{f} | w, k_\theta \sim \mathcal{N}(\mathbf{f} | m(X; w), K(X, X))$

likelihood: $Y | \mathbf{f}, \sigma^2 \sim \mathcal{N}(Y | \mathbf{f}, \sigma^2 \mathbf{I})$. (11)

For notational simplicity, the dependence on inputs is ignored. At new testing inputs X_* , the predictive distribution of outputs Y_* is:

$$p(Y_* | Y) = \int p(Y_* | \mathbf{f}, w, k_\theta, \sigma^2) p(\mathbf{f}, w, k_\theta, \sigma^2 | Y) d\mathbf{f} dw dk_\theta d\sigma^2. \quad (12)$$

The joint posterior $p(\mathbf{f}, w, k_\theta, \sigma^2 | Y)$ of the parameter $\mathbf{f}$ and hyperparameters $\{w, k_\theta, \sigma^2\}$ can be factorized as the product of the conditional posterior of $\mathbf{f}$ and the marginal posterior of $\{w, k_\theta, \sigma^2\}$:

$$p(\mathbf{f}, w, k_\theta, \sigma^2 | Y) = p(\mathbf{f} | w, k_\theta, \sigma^2, Y) p(w, k_\theta, \sigma^2 | Y). \quad (13)$$

After substituting Eq. (13) into Eq. (12) and rearranging sequence of integrands, the predictive distribution can be rewritten as:

$$\begin{aligned} p(Y_* | Y) &= \int p(Y_* | \mathbf{f}, w, k_\theta, \sigma^2) p(\mathbf{f} | w, k_\theta, \sigma^2, Y) \\ &\quad p(w, k_\theta, \sigma^2 | Y) d\mathbf{f} dw dk_\theta d\sigma^2 \\ &= \int p(w, k_\theta, \sigma^2 | Y) \left\{ \int p(Y_* | \mathbf{f}, w, k_\theta, \sigma^2) \right. \\ &\quad \left. p(\mathbf{f} | w, k_\theta, \sigma^2, Y) d\mathbf{f} \right\} dw dk_\theta d\sigma^2. \end{aligned} \quad (14)$$

The inner integral $\int p(Y_* | \mathbf{f}, w, k_\theta, \sigma^2) p(\mathbf{f} | w, k_\theta, \sigma^2, Y) d\mathbf{f}$ reduces to the predictive distribution under the standard GPR with hyperparameters fixed, which is a Gaussian distribution:

$$\begin{aligned} &\int p(Y_* | \mathbf{f}, w, k_\theta, \sigma^2) p(\mathbf{f} | w, k_\theta, \sigma^2, Y) d\mathbf{f} \\ &= p(Y_* | Y, w, k_\theta, \sigma^2) = \mathcal{N}(Y_* | \mu_*, \Sigma_*), \text{ where} \\ \mu_* &= m(X_*; w) + K(X_*, X) [K(X, X) + \sigma^2 \mathbf{I}]^{-1} (Y - m(X; w)) \\ \Sigma_* &= K(X_*, X_*) + \sigma^2 \mathbf{I} - K(X_*, X) [K(X, X) + \sigma^2 \mathbf{I}]^{-1} K(X, X_*). \end{aligned} \quad (15)$$

After substituting Eq. (15) into Eq. (14), the predictive distribution becomes:

$$\begin{aligned} p(Y_* | Y) &= \int p(w, k_\theta, \sigma^2 | Y) p(Y_* | Y, w, k_\theta, \sigma^2) dw dk_\theta d\sigma^2 \\ &= \mathbb{E}_{p(w, k_\theta, \sigma^2 | Y)} [p(Y_* | Y, w, k_\theta, \sigma^2)] \\ &\approx \frac{1}{M} \sum_{m=1}^M p(Y_* | Y, \{w, k_\theta, \sigma^2\}^m) \\ &= \frac{1}{M} \sum_{m=1}^M \mathcal{N}\left(Y_* \middle| \mu_*^{\{w, k_\theta, \sigma^2\}^m}, \Sigma_*^{\{w, k_\theta, \sigma^2\}^m}\right) \end{aligned} \quad (16)$$

where $\mu_{*}^{[w, k_{\theta}, \sigma^2]^m}$ and $\Sigma_{*}^{[w, k_{\theta}, \sigma^2]^m}$ are the predictive mean and variance in Eq. (15) with hyperparameters fixed as $\{w, k_{\theta}, \sigma^2\}^m$. The second line is the expectation with respect to the posterior of hyperparameters $p(w, k_{\theta}, \sigma^2|Y)$. In the third line, the Monte Carlo integration is used to approximate the expectation, with M samples $\{w, k_{\theta}, \sigma^2\}^m$ ($m = 1, 2, \dots, M$) drawn from $p(w, k_{\theta}, \sigma^2|Y)$. Finally the predictive distribution at testing inputs X_* is approximated as a mixture of Gaussian, as shown in the last line of Eq. (16).

The remaining job is to draw samples from the posterior of hyperparameters $p(w, k_{\theta}, \sigma^2|Y)$, which can be derived from the Bayes theorem over hyperparameters $\{w, k_{\theta}, \sigma^2\}$:

$$p(w, k_{\theta}, \sigma^2|Y) \propto p(Y|w, k_{\theta}, \sigma^2)p(w, k_{\theta}, \sigma^2). \quad (17)$$

Sampling from $p(w, k_{\theta}, \sigma^2|Y)$ is equivalent to sampling from $p(Y|w, k_{\theta}, \sigma^2)p(w, k_{\theta}, \sigma^2)$ [16], where $p(w, k_{\theta}, \sigma^2)$ is the assigned prior. And $p(Y|w, k_{\theta}, \sigma^2)$ can be derived according to the Bayes theorem over the parameter $\mathbf{f}$:

$$p(\mathbf{f}|Y, w, k_{\theta}, \sigma^2) = \frac{p(Y|\mathbf{f}, \sigma^2)p(\mathbf{f}|w, k_{\theta})}{p(Y|w, k_{\theta}, \sigma^2)} \quad (18)$$

$p(Y|w, k_{\theta}, \sigma^2)$ is the denominator calculated by:

$$\begin{aligned} p(Y|w, k_{\theta}, \sigma^2) &= \int p(Y|\mathbf{f}, \sigma^2)p(\mathbf{f}|w, \sigma^2)d\mathbf{f} \\ &= \mathcal{N}(Y|m(X; w), k(X, X) + \sigma^2\mathbf{I}) \end{aligned} \quad (19)$$

Finally the Markov Chain Monte Carlo algorithm can be used to draw samples from $p(Y|w, k_{\theta}, \sigma^2)p(w, k_{\theta}, \sigma^2)$.

4. Simulations

4.1 Cart-Pole System

The cart-pole system as shown in Fig. 1 is chosen for simulation, which is implemented by Box2D, an open-source, freely available 2D physics simulator. The system consists of a cart with mass $m_1 = 0.5\text{kg}$ and a pole with mass $m_2 = 0.5\text{kg}$ and length $l = 0.6\text{m}$. The time discretization is $\Delta t = 0.1\text{s}$. The cart moves horizontally with an applied external force u . During each sampling period of $\Delta t = 0.1\text{s}$, the force u takes a constant value, which is generated from a uniform distribution on the interval $[-15, 15]\text{N}$. The maximum friction between the cart and its rail is 3N . The pole angle θ is measured anticlockwise from hanging down.

The cart-pole system has 4 states: the position of the cart x , its velocity $\dot{x}$, the angle of the pole θ , and the angular velocity $\dot{\theta}$, which are collected as $z = [z_1, z_2, z_3, z_4]^T = [x, \dot{x}, \theta, \dot{\theta}]^T$. Both system noise v and measurement noise w are included:

Fig. 1 Cart-pole system.

$$\begin{aligned} z_{k+1} &= f(z_k, u_k) + v_k \\ y_k &= z_k + w_k. \end{aligned} \quad (20)$$

For each state, the system noise is a zero-mean Gaussian distribution. For states $[x, \dot{x}, \theta, \dot{\theta}]$, the corresponding variances of system noise are $[0, 0.001, 0, 0.001]$. The measurement noise with a zero-mean Gaussian distribution is added to each state. For states $[x, \dot{x}, \theta, \dot{\theta}]$, the corresponding variances of measurement noise are $[0, 0.01, 0.005, 0.02]$. There is no measurement noise for position x because the position of the cart is less important for the cart-pole system's dynamics.

Assume that the analytic ODEs derived from the Lagrange formalism are frictionless:

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \\ \dot{z}_4 \end{bmatrix} = \begin{bmatrix} z_2 \\ \frac{2m_2 l z_4^2 \sin z_3 + 3m_2 g \sin z_3 \cos z_3 + 4u}{4(m_1 + m_2) - 3m_2 \cos^2 z_3} \\ z_4 \\ \frac{-3m_2 l z_4^2 \sin z_3 \cos z_3 - 6(m_1 + m_2)g \sin z_3 - 6u \cos z_3}{4l(m_1 + m_2) - 3m_2 l \cos^2 z_3} \end{bmatrix}. \quad (21)$$

Therefore there is an error between the true dynamic system and the analytic ODEs.

Initially the system stays static and the pole hangs down. Starting from the initial state $z_1 = [0, 0, 0, 0]^T$, applying a random force u at each time step, a trajectory containing $N = 100$ samples $(X, Y) = (x_i, y_i)_{i=1}^N$ are collected as training data. Each input x_i includes the current state z_i and the taken action u_i ; the corresponding output y_i is the next state z_{i+1} . Then another testing trajectory containing $N_* = 200$ samples is generated similarly.

4.2 Cart-Pole System with Highly Nonlinear Friction

Another cart-pole system realized by Box2D with more complicated dynamics is considered. It has the same physical parameters (cart mass m_1 , pole mass m_2 and pole length l), the same time discretization Δt as the dynamic system in Sect. 4.1. The maximum friction between the cart and its rail is 3N . The friction torque between the cart and the pole is:

$$\tau = -[0.5 \sin \theta + 0.2 \sin 2\theta] \text{sgn}(\dot{\theta}) (\text{Nm}), \quad \theta \in [0, 2\pi]. \quad (22)$$

During each sampling period of $\Delta t = 0.1\text{s}$, the force u takes a constant value, which is generated from a uniform distribution on the interval $[-20, 20]\text{N}$.

For each state, the system noise is a zero-mean Gaussian distribution. For states $[x, \dot{x}, \theta, \dot{\theta}]$, the corresponding variances of system noise are $[0, 0.0005, 0, 0.0005]$. The measurement noise with a zero-mean Gaussian distribution is added to each state. For states $[x, \dot{x}, \theta, \dot{\theta}]$, the corresponding variances of measurement noise are $[0, 0.01, 0.005, 0.01]$.

Assume that the analytic ODEs derived from the Lagrange formalism are frictionless, and given by Eq. (21). Obviously compared with the dynamic system in Sect. 4.1, the error between the cart-pole system with highly nonlinear friction and the analytic ODEs is much larger. Thus the analytic ODEs become much more inaccurate. Starting from

the initial state $z_1 = [0, 0, 0, 0]^T$, applying a random force u at each time step, a trajectory containing $N = 200$ samples $(X, Y) = (x_i, y_i)_{i=1}^N$ are collected as training data. Then another testing trajectory containing $N_* = 200$ samples is generated similarly.

To verify the benefit of the proposed method, three baselines are chosen: 1) the standard Gaussian process regression; 2) the ODE method; 3) the method in [8] as shown in Sect. 3.1. For the proposed method and the method in [8], the one-time-step integration of analytic ODEs in Eq. (21) is used as the mean function of the Gaussian process prior. For the standard Gaussian process regression, the information provided by analytic ODEs is discarded and the system function is modelled by a zero-mean Gaussian process whose hyperparameters are learned by evidence maximization. For the ODE method, at first analytic ODEs are fit to training data by least square, which can be realized by Eq. (7). Then analytic ODEs with the optimized physical parameters are used for prediction. For the ODE method, only the point estimate of the prediction can be obtained. In the following discussion, "GP+ODE proposed" refers to the proposed method; "GP+ODE existing" refers to the method in [8]; "standard GP" refers to the standard Gaussian process regression; "ODE" refers to the ODE method.

Because the observed value of angle θ belongs to $[0, 2\pi]$, it is a discontinuous function of the input (namely state and action). It is hard for the GPR to model discontinuity because of the assumption of the kernel function: if inputs are similar (close) to each other, outputs will be strongly correlated. The difficulty of modelling discontinuity can be avoided by extending the angle θ to $(\sin \theta, \cos \theta)$. Then both $\sin \theta$ and $\cos \theta$ are continuous functions of the input. The state becomes $z = [x, \dot{x}, \sin \theta, \cos \theta, \dot{\theta}]^T$ with 5 dimensions. Hence the output of the regression problem has 5 dimensions. An independent GPR is used for each output dimension.

For the proposed method (GP+ODE proposed), all hyperparameters $\{w, k_\theta, \sigma^2\}$ are assigned a Gamma prior: $w, k_\theta, \sigma^2 \sim \text{Gamma}(1, 1)$. Since it is difficult to obtain the gradient with respect to the physical parameter w , instead of Hamiltonian Monte Carlo, Metropolis-Hastings algorithm is used to draw samples from the posterior of hyperparameters $p(w, k_\theta, \sigma^2 | Y)$.

5. Results and Discussion

5.1 Cart-Pole System

At first, for each method the training data is used to learn the corresponding parameters. Then the predictive perfor-

mances are compared in terms of the one-time-step prediction and the long-term prediction on the testing data. The training time for each method is shown in Table 1.

5.1.1 One-Time-Step Prediction

Given the testing data $(x_j^*, y_j^*)_{j=1}^{N_*} = (X^*, Y^*)$, using GP+ODE proposed, GP+ODE existing and standard GP, the predictive distributions at testing inputs $\{x_1^*, x_2^*, \dots, x_{N_*}^*\}$ are obtained; using ODE, the point estimate of the prediction is obtained. The predictive performance can be measured by the predictive accuracy and the predictive uncertainty. The predictive accuracy can be quantified by the root mean square error (RMSE):

$$\text{RMSE} = \sqrt{\frac{1}{N_*} \sum_{j=1}^{N_*} (\bar{y}_j - y_j^*)^2}, \quad (23)$$

where $\bar{y}_j$ is the predictive mean at the j th testing input; y_j^* is the corresponding true output value. The predictive uncertainty can be quantified by the mean standardized log loss (MSLL), which is the mean negative log probability of testing data under the learned model $\mathcal{M}$ [4]:

$$\text{MSLL} = \frac{1}{N_*} \sum_{j=1}^{N_*} \{-\log p(y_j^* | x_j^*, X, Y, \mathcal{M})\}, \quad (24)$$

where $p(\cdot)$ is the predictive distribution under the learned model $\mathcal{M}$. The lower values of RMSE and MSLL indicate the better predictive performance. The RMSE and MSLL for different methods are summarized in Tables 2 and 3. Since ODE can only obtain the point estimation of the prediction, its predictive uncertainty is not available in Table 3. Each column corresponds to each component of the state; each row corresponds to one kind of methods. The last column of Table 2 is the RMSE of the full state $z = [x, \dot{x}, \sin \theta, \cos \theta, \dot{\theta}]^T$, which is calculated by $\sqrt{1/N_* \sum_{j=1}^{N_*} \|\bar{z}_j - z_j^*\|^2}$, with $\bar{z}_j$ the predictive mean, z_j^* the true value.

As shown in Table 2, GP+ODE proposed has the smallest RMSE; GP+ODE existing has the smaller RMSE than standard GP and ODE. As shown in Table 3, GP+ODE proposed has the smallest MSLL; GP+ODE existing and standard GP have similar MSLL. To sum up, GP+ODE proposed has the best predictive performance. The reasons are

Table 1 Training times (s)

GP+ODE proposed	GP+ODE existing	standard GP	ODE
1432.8	2.3	0.5	1.5

Table 2 Predictive accuracy: RMSE

methods	x	$\dot{x}$	$\sin \theta$	$\cos \theta$	$\dot{\theta}$	full state
GP+ODE proposed	0.013	0.227	0.082	0.074	0.437	0.505
GP+ODE existing	0.013	0.274	0.079	0.079	0.481	0.565
standard GP	0.039	0.327	0.127	0.142	0.533	0.655
ODE	0.028	0.367	0.080	0.077	0.698	0.797

as follows. At first, ODE has the largest RMSE because the analytic ODEs are inaccurate relative to the true dynamic system. Then, GP+ODE proposed and GP+ODE existing are able to combine the benefits of GPR and analytic ODEs. In areas with little training data, standard GP has bad predictive performance because of the poor extrapolation ability of GPR. In contrast, GP+ODE proposed and GP+ODE existing could use the information of analytic ODEs to improve the accuracy of the learned model in areas with little training data. In areas with enough training data, GP+ODE proposed and GP+ODE existing could model the residual between the true dynamic system and analytic ODEs accurately because of the high flexibility of GPR. Finally, GP+ODE existing leads to a bad suboptimal model since physical parameters of analytic ODEs and parameters of the kernel function, the noise variance are optimized separately. That is why GP+ODE proposed has the best predictive performance.

5.1.2 Long-Term Prediction

In this subsection the n -steps-ahead predictive performances of different methods are compared. For the testing trajectory $(x_j^*, y_j^*)_{j=1}^{N_s} = (X^*, Y^*)$, each input x_j^* includes the current state z_j^* and the taken action u_j^* ; each output y_j^* is the consecutive state z_{j+1}^* . Starting from the same initial state z_k^* , taking

the same action sequence $\{u_k^*, u_{k+1}^*, \dots, u_{k+n-1}^*\}$, 4 methods are used to make the n -steps-ahead prediction.

For the long-term prediction, the correlation between consecutive predicted data points should be considered [8]. To do so, the future data point should depend on data points predicted so far, which can be realized by adding already predicted data points to training data (X, Y) :

$$\begin{aligned} \bar{y}_k &\sim p(\bar{y}_k | \{(X, Y)\}, \bar{x}_k), \quad \bar{x}_k = (z_k^*, u_k^*) \\ \bar{y}_{k+1} &\sim p(\bar{y}_{k+1} | \{(X, Y), (\bar{x}_k, \bar{y}_k)\}, \bar{x}_{k+1}), \\ \bar{x}_{k+1} &= (\bar{y}_k, u_{k+1}^*) \\ &\vdots \\ \bar{y}_{k+n-1} &\sim p(\bar{y}_{k+n-1} | \{(X, Y), (\bar{x}_k, \bar{y}_k), \dots, (\bar{x}_{k+n-2}, \bar{y}_{k+n-2})\}, \\ &\quad \bar{x}_{k+n-1}), \quad \bar{x}_{k+n-1} = (\bar{y}_{k+n-2}, u_{k+n-1}^*). \end{aligned} \quad (25)$$

Here the generated predictive trajectory is denoted as: $\{\bar{y}_k, \bar{y}_{k+1}, \dots, \bar{y}_{k+n-1}\}$, which is compared with the true trajectory: $\{y_k^*, y_{k+1}^*, \dots, y_{k+n-1}^*\}$.

Two situations are considered: 1) standard GP has the poor one-time-step predictive performance, such as the testing data around 4.9s, as shown in Fig. 2; 2) ODE has the poor one-time-step predictive performance, such as the testing data around 14.2s, as shown in Fig. 3. To clearly show the poor predictive performance of ODE, the predictive standard deviations are not shown in Fig. 3. Choosing the states at 4.9s and 14.2s as initial states respectively, 10-time-step predictive trajectories are generated, as illustrated in Figs. 4 and 5.

As shown in Fig. 4, with the initial state given by the state at 4.9s, the predictive trajectories obtained from stan-

Table 3 Predictive uncertainty: MSLL

methods	x	$\dot{x}$	$\sin \theta$	$\cos \theta$	θ
GP+ODE proposed	-2.87	-0.06	-1.08	-1.19	0.61
GP+ODE existing	-2.81	0.19	-1.09	-1.04	0.83
standard GP	-1.91	0.18	-0.90	-0.99	0.78

Fig. 2 Cart-pole system: comparisons of predictive distributions for one-time-step prediction. Shaded regions represent 2 standard deviations. (testing data during 0.7s-5.5s)

Fig. 3 Cart-pole system: comparisons of predictive means for one-time-step prediction. (testing data during 10.5s-15s)

Fig. 4 Cart-pole system: 10-time-step predictive trajectories with the initial state given by the state at 4.9s. Predictive step 0 denotes the initial state.

Fig. 5 Cart-pole system: 10-time-step predictive trajectories with the initial state given by the state at 14.2s.

standard GP deviates from the true trajectories seriously; the predictive trajectories obtained from GP+ODE proposed follow the true trajectories best. The reasons are as follows.

In Fig. 2, for testing data around 4.9s, standard GP has much higher predictive uncertainty, which means that there is little training data in these areas. Therefore standard GP has the

bad long-term predictive performance because of the poor extrapolation ability of GPR. In contrast, GP+ODE proposed and GP+ODE existing use the one-time-step integration of analytic ODEs as the mean function of the Gaussian process prior. In areas with little training data, the predictive mean of GPR will decay to the mean function of the Gaussian process prior, namely the one-time-step integration of analytic ODEs. Hence GP+ODE proposed and GP+ODE existing could make use of the information of analytic ODEs to improve the accuracy of the learned model in areas with little training data. Since GP+ODE existing leads to a bad suboptimal model (as shown in Tables 2 and 3), GP+ODE proposed has the best long-term predictive performance in Fig. 4.

As shown in Fig. 5, with the initial state given by the state at 14.2s, the predictive trajectories obtained from ODE deviates from the true trajectories seriously; the predictive trajectories obtained from GP+ODE proposed follow the true trajectories best. The reason why ODE behaves worst is that analytic ODEs are highly inaccurate in these areas, as shown in Fig. 3. In contrast, there is enough training data in these areas because the one-time-step prediction of standard GP is relatively accurate. As a result, GP+ODE proposed and GP+ODE existing could model the residual between the true dynamic system and analytic ODEs accurately because of the high flexibility of GPR. Since GP+ODE existing leads to a bad suboptimal model, GP+ODE proposed has the best long-term predictive performance in Fig. 5.

For the cart-pole system, GP+ODE proposed has the best predictive performance, but with much longer training time. GP+ODE existing has the worse predictive performance because it learns a bad suboptimal model.

5.2 Cart-Pole System with Highly Nonlinear Friction

At first, for each method the training data is used to learn the corresponding parameters. Then the predictive performance is compared in terms of the one-time-step prediction and the long-term prediction on the testing data. The training time for each method is shown in Table 4.

5.2.1 One-Time-Step Prediction

On the testing data $(x_j^*, y_j^*)_{j=1}^{N_s} = (X^*, Y^*)$, the predictive performance is compared in terms of the predictive accuracy

Table 4 Training times (s)

GP+ODE proposed	GP+ODE existing	standard GP	ODE
2933.8	5.4	2.5	2.8

and the predictive uncertainty. The predictive accuracy is quantified by the RMSE in Eq. (23). The predictive uncertainty is quantified by MSLL in Eq. (24). The RMSE and MSLL for different methods are summarized in Tables 5 and 6.

From Table 5, GP+ODE proposed has the smallest RMSE; GP+ODE existing has the smaller RMSE than standard GP and ODE. From Table 6, GP+ODE proposed has the smallest MSLL; standard GP has smaller MSLL than GP+ODE existing. To sum up, GP+ODE proposed has the best predictive performance. The reasons are similar to those in Sect. 5.1.1. At first, ODE has the largest RMSE because the analytic ODEs are highly inaccurate relative to the true dynamic system. Then, GP+ODE proposed and GP+ODE existing could combine the benefits of GPR and analytic ODEs. In areas with little training data, standard GP has bad predictive performance because of the poor extrapolation ability of GPR. In contrast, GP+ODE proposed and GP+ODE existing will make use of the information of analytic ODEs to improve the accuracy of the learned model in the areas with little training data. In areas with enough training data, GP+ODE proposed and GP+ODE existing could model the residual between the true dynamic system and analytic ODEs accurately because of the high flexibility of GPR. Finally, GP+ODE existing leads to a bad suboptimal model since physical parameters of analytic ODEs and parameters of the kernel function, the noise variance are optimized separately. That is why GP+ODE proposed has the best predictive performance.

5.2.2 Long-Term Prediction

The n -steps-ahead predictive performances of different methods are compared. Two situations are considered: 1) standard GP has the poor one-time-step predictive performance; 2) ODE has the poor one-time-step predictive performance. It is found that ODE has the poor one-time-step predictive performance for testing data around 5.7s, and standard GP has the poor one-time-step predictive performance for testing data around 8.7s. Choosing the states at 5.7s and 8.7s as initial states respectively, 10-time-step predictive trajectories are generated, as illustrated in Figs. 6 and 7.

As shown in Fig. 6, with the initial state given by the

Table 6 Predictive uncertainty: MSLL

methods	x	$\dot{x}$	$\sin \theta$	$\cos \theta$	θ
GP+ODE proposed	-3.05	-0.37	-1.11	-1.25	0.69
GP+ODE existing	-3.05	0.01	-1.14	-1.21	1.33
standard GP	-2.35	-0.09	-1.07	-1.24	0.93

Table 5 Predictive accuracy: RMSE

methods	x	$\dot{x}$	$\sin \theta$	$\cos \theta$	θ	full state
GP+ODE proposed	0.012	0.167	0.079	0.068	0.500	0.538
GP+ODE existing	0.012	0.221	0.077	0.072	0.607	0.655
standard GP	0.022	0.228	0.084	0.071	0.654	0.701
ODE	0.021	0.351	0.079	0.072	0.931	1.001

Fig. 6 Cart-pole system with highly nonlinear friction: 10-time-step predictive trajectories with the initial state given by the state at 5.7s.

Fig. 7 Cart-pole system with highly nonlinear friction: 10-time-step predictive trajectories with the initial state given by the state at 8.7s.

state at 5.7s, the predictive trajectories obtained from ODE deviates from the true trajectories seriously; the predictive trajectories obtained from GP+ODE proposed follow the true trajectories best. As shown in Fig. 7, with the initial state given by the state at 8.7s, the predictive trajectories obtained from standard GP deviates from the true trajectories seriously; the predictive trajectories obtained from GP+ODE proposed follow the true trajectories best. The reasons have been analyzed in Sect. 5.2.1. GP+ODE proposed and GP+ODE existing combine the benefits of GPR and analytic ODEs. In areas with little training data, GP+ODE proposed and GP+ODE existing use the information of analytic ODEs to improve the accuracy of the learned model; in areas with enough training data, GP+ODE proposed and GP+ODE existing model the residual between the true dynamic system and analytic ODEs accurately because of the high flexibility of GPR. Since GP+ODE existing leads to a bad suboptimal model, GP+ODE proposed has the best long-term predictive performance.

For the cart-pole system with the highly nonlinear friction, GP+ODE proposed has the best predictive performance, but with much longer training time. GP+ODE existing has the worse predictive performance because it leads to a bad suboptimal model.

The above two simulations show the benefit of com-

bing analytic ODEs and GPR. GP+ODE existing has very short training time, but there is the risk of learning a sub-optimal model. GP+ODE proposed is a gradient-free optimization method, and is more promising to learn an optimal model since all parameters are optimized simultaneously. However, GP+ODE proposed has the limitation of a long training time.

6. Conclusion

The paper proposes a method to learn dynamic systems. As prior information, the analytic ODEs of the dynamic system is incorporated into the GPR learning framework. By this way, the high flexibility of the GPR and the global generalization ability of analytic ODEs can be combined. A novel method is proposed to simultaneously learn all parameters, which is realized by the fully Bayesian GPR. The standard Gaussian process regression, the ODE method and the existing method in the literature are chosen as baselines to verify the advantage of the proposed method. The simulation results show that the proposed method has the competitive predictive performance.

In future work, the proposed method will be used to learn the real dynamic systems. For systems where the analytic ODEs are difficult to be derived, the corresponding simulators can be built and then used as the mean function

of the Gaussian process prior. The unknown parameters of the mean function are the simulator's physical parameters. Furthermore, the long training time of the proposed method will be considered. On the one hand, more efficient Markov Chain Monte Carlo algorithms will be considered. On the other hand, a balance can be taken between the proposed method and the method in [8]. The total parameters can be optimized iteratively. At first physical parameters of the mean function are optimized; then parameters of the kernel function, the noise variance are optimized with physical parameters fixed as the optimized values. The two procedures are repeated until convergence.

References

- [1] G. Pillonetto and G.D. Nicolao, "A new kernel-based approach for linear system identification," *Automatica*, vol.46, no.1, pp.81–93, Jan. 2010.
- [2] S. Eleftheriadis, T.F.W. Nicholson, M.P. Deisenroth, and J. Hensman, "Identification of Gaussian process state space models," *Advances in Neural Information Processing Systems*, pp.5309–5319, 2017.
- [3] K. Fujimoto, A. Taniguchi, and Y. Nishida, "System identification of nonlinear state-space models with linearly dependent unknown parameters based on variational bayes," *SICE J. Control, Measurement, and System Integration*, vol.11, no.6, pp.456–462, Jan. 2018.
- [4] C.E. Rasmussen and C.K.I. Williams, *Gaussian processes for machine learning*, The MIT Press, 2005.
- [5] K. Fujimoto, H. Beppu, and Y. Takaki, "Numerical solutions of Hamilton-Jacobi inequalities by constrained Gaussian process regression," *SICE J. Control, Measurement, and System Integration*, vol.11, no.5, pp.419–428, Jan. 2018.
- [6] C.E. Rasmussen and M. Kuss, "Gaussian processes in reinforcement learning," *Seventeenth Annual Conference on Neural Information Processing Systems (NIPS 2003)*, MIT Press, pp.751–758, 2004.
- [7] M.P. Deisenroth, D. Fox, and C.E. Rasmussen, "Gaussian processes for data-efficient learning in robotics and control," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol.37, no.2, pp.408–423, Feb. 2013.
- [8] J. Ko, D.J. Klein, D. Fox, and D. Haehnel, "Gaussian processes and reinforcement learning for identification and control of an autonomous blimp," *Proc. 2007 IEEE Int. Conf. Robot. Autom.*, pp.742–747, 2007.
- [9] T. Wu and J. Movellan, "Semi-parametric Gaussian process for robot system identification," *2012 IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, pp.725–731, 2012.
- [10] D. Romeres, D.K. Jha, A. DallaLibera, B. Yezazunis, and D. Nikovski, "Semiparametrical Gaussian processes learning of forward dynamical models for navigating in a circular maze," *2019 Int. Conf. Robot. Autom. (ICRA)*, IEEE, pp.3195–3202, 2019.
- [11] M. Cutler and J.P. How, "Efficient reinforcement learning for robots using informative simulated priors," *2015 IEEE Int. Conf. Robot. Autom. (ICRA)*, pp.2605–2612, 2015.
- [12] D. Nguyen-Tuong and J. Peters, "Using model knowledge for learning inverse dynamics," *2010 IEEE Int. Conf. Robot. Autom.*, pp.2677–2682, 2010.
- [13] D. Romeres, M. Zorzi, R. Camoriano, and A. Chiuso, "Online semi-parametric learning for inverse dynamics modeling," *2016 IEEE 55th Conf. Decision and Control (CDC)*, pp.2945–2950, 2016.
- [14] E.M. Schearer, Y.-W. Liao, E.J. Perreault, M.C. Tresch, W.D. Memberg, R.F. Kirsch, and K.M. Lynch, "Semiparametric identification of human arm dynamics for flexible control of a functional electrical stimulation neuroprosthesis," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol.24, no.12, pp.1405–1415, Dec. 2016.
- [15] R. Sebastian and F. Stulp, "Comparing semi-parametric model learning algorithms for dynamic model estimation in robotics," *arXiv preprint arXiv:1906.11909*, 2019.

- [16] C.M. Bishop, *Pattern recognition and machine learning*, Springer, 2006.

Shengbing Tang is a Ph.D. student in the Department of Aeronautics and Astronautics, Kyoto University. His research interests are Bayesian statistics and machine learning.

Kenji Fujimoto received his B.Sc. and M.Sc. degrees in Engineering and Ph.D. degree in Informatics from Kyoto University, Japan, in 1994, 1996 and 2001, respectively. He is currently a professor of Graduate School of Engineering, Kyoto University, Japan. From 1997 to 2004 he was a research associate of Graduate School of Engineering and Graduate School of Informatics, Kyoto University, Japan. From 2004 to 2012 he was an associate professor of Graduate School of Engineering, Nagoya University, Japan. His research interests include nonlinear control and stochastic systems theory.

Ichiro Maruta received the Bachelor of Engineering, Master of Informatics, and Doctor of Informatics degrees from Kyoto University in 2006, 2008, and 2011, respectively. He was a research fellow of the Japan Society for the Promotion of Science from 2008 to 2011. From 2012 to 2017, he was an Assistant Professor at the Graduate School of Informatics, Kyoto University. In 2017, he joined the Graduate School of Engineering, Kyoto University, as a Lecturer of the Department of Aeronautics and Astronautics, and since 2019, he is an Associate Professor.