Learning of Hamiltonian Dynamics with Reproducing Kernel Hilbert Spaces

Torbjørn Smith and Olav Egeland, *Senior Member, IEEE*

Abstract— This paper presents a method for learning Hamiltonian dynamics from a limited set of data points. The Hamiltonian vector field is found by regularized optimization over a reproducing kernel Hilbert space of vector fields that are inherently Hamiltonian, and where the vector field is required to be odd or even. This is done with a symplectic kernel, and it is shown how this symplectic kernel can be modified to be odd or even. The performance of the method is validated in simulations for two Hamiltonian systems. The simulations show that the learned dynamics reflect the energy-preservation of the Hamiltonian dynamics, and that the restriction to symplectic and odd dynamics gives improved accuracy over a large domain of the phase space.

I. INTRODUCTION

Data-driven techniques have been shown to be powerful in system identification of dynamical systems. When a set of measurements is given, machine learning techniques can be used to identify the dynamics of the underlying system [1]. The performance of data-driven methods depends on the quality of the data that is used for learning. Data-driven methods may fail to generalize beyond the given data set [2], and may suffer from overfitting if the data set is limited or noisy [3]. In practice it can be can be labor-intensive to assemble a viable data set and this can be impractical in applications. As the data set grows, the computational cost of learning the model increases, and the inference time of the final learned model can be high [2] [4]. Producing stable and robust models for safety-critical and control applications is also vital [5]. To deal with these challenges, researchers have deployed various methods to improve or constrain the learning of dynamical systems using priors, which can lead to good results even when the data set is limited [3] [4].

Related work: Price prediction for financial futures contracts was studied in [6] using a data-driven approach with functions in a reproducing kernel Hilbert space (RKHS). Odd symmetric price action was assumed, and a reproducing kernel was proposed so that the learned functions in the RKHS satisfied this constraint. The odd symmetry constraint improved prediction and reduced overfitting compared to an unconstrained implementation.

Kernel-based methods with an RKHS were used in [7] to learn the inverse dynamics of a manipulator by learning the Lagrangian of the system. Polynomial basis functions were used in [8] for control-oriented learning of Lagrangian and Hamiltonian functions from trajectory data. This allowed for

accurate and generalized learning from a limited number of trajectories. The *learning-from-demonstrations* problem was addressed in [2], where a dynamical system was learned to copy human demonstrations. Human-drawn shapes were imitated using a data-driven approach, and a dynamical system with the desired equilibrium points was found using an RKHS formulation with random Fourier features for dimensionality reduction. Point-wise contraction constraints were enforced along the trajectories to create a contraction region around the desired trajectory and to condition the learned vector field.

The Hamiltonian dynamics of energy-conserving systems were learned in [9] by learning the Hamiltonian function using neural nets. This significantly improved the predictive accuracy of the learned system. This work was further developed in [10], where the need for higher-order derivatives of the generalized coordinates was eliminated, and the option for energy-based control was included. In [11], the work in [9] was further developed by using the symplectic Leapfrog integrator to integrate the partial derivatives of the learned Hamiltonian and by back-propagating the loss through the integrator over multiple time steps. This improved the learning of more complex and noisy Hamiltonian systems.

The learning of nonlinear dynamics with stabilizability as a side constraint was investigated in [4]. The nonlinear dynamics were learned under a contraction constraint and the method was validated using a planar drone model. The contraction constraint improved the learned trajectory generation and tracking while making the learning process more data efficient. The model was learned in an RKHS, and random Fourier features were used for dimensionality reduction. Nonlinear system identification using a data-driven approach was investigated in [12] by including constraints that enforce prior knowledge of the region of attraction. The stability region was enforced using a Lyapunov function, and the hypothesis space for the learned system was an RKHS. In [13], they learned dynamical systems with prior knowledge in an RKHS, with the addition of a bias term in the regularized least squares cost. The bias term was used to embed prior knowledge of the underlying system to aid the learning process, improving data efficiency and out-of-sample generalization.

A method for optimization-based learning of vector fields with side constraints from a limited number of trajectories was presented in [3]. Polynomial basis functions were used, and it was shown how to include side constraints in semidefinite programming, which improved learning accuracy. Side constraints for the vector fields included interpolation at a

The authors are with the Department of Mechanical and Industrial Engineering, The Norwegian University of Science and Technology (NTNU), NO-7491 Trondheim, Norway. (email: torbjorn.smith@ntnu.no; olav.egeland@ntnu.no)

finite set of points, sign symmetry, gradient and Hamiltonian dynamics, coordinate non-negativity, directional monotonicity, and invariance of sets.

Contribution: In this paper, we show how Hamiltonian dynamical systems with odd vector fields can be learned in an RKHS setting by selecting a kernel that ensures that the learned vector fields are Hamiltonian and odd symmetric. By learning the dynamical system with the Hamiltonian formalism, energy conservation is enforced in the model [9]. The odd symmetry constraint further prunes the hypothesis space to improve the predictive performance of the learned model across the domain of the learned model [6]. Encoding the constraints in the kernel improves learning time as the straightforward closed-form solution of the learning problem is retained. It is shown with two examples that the generalization properties of the learned model when using out-of-sample data points are greatly improved through the additional constraints.

The paper is organized as follows: Section II presents the problem investigated in this work. Section III reviews the relevant theory for reproducing kernel Hilbert spaces, learning dynamical systems, and Hamiltonian mechanics. The proposed method is presented in Section IV as an odd symplectic kernel is developed. Section V presents the numerical experiments used to verify the proposed method. Finally, Section VI presents the conclusion and future work.

II. PROBLEM FORMULATION

The problem that is investigated in this paper is how to learn the Hamiltonian dynamics of an unknown system from a limited set of data. The system dynamics are given by

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) \tag{1}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ is the state vector, $\dot{\boldsymbol{x}} \in \mathbb{R}^n$ is the time derivative of the state vector, and $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^n$ are the system dynamics. It is assumed that $\boldsymbol{y} = \dot{\boldsymbol{x}}$ is available as a measurement or from numerical differentiation. Given a set of N data points $\{(\boldsymbol{x}_i, \boldsymbol{y}_i) \in \mathbb{R}^n \times \mathbb{R}^n\}_{i=1}^N$ from simulations or measurements, the aim is to learn a function $\boldsymbol{f}^* \in \mathcal{F}$, where the class of functions $\mathcal{F}$ is a reproducing kernel Hilbert space (RKHS) with inner product $\langle \cdot, \cdot \rangle_{\mathcal{F}}$ and norm $\|\boldsymbol{f}\|_{\mathcal{F}}^2 = \langle \boldsymbol{f}, \boldsymbol{f} \rangle_{\mathcal{F}}$ [14]. In that case the function $\boldsymbol{f}^*$ is found by the regularized minimization problem [15]

$$\boldsymbol{f}^* = \arg\min_{\boldsymbol{f} \in \mathcal{F}} \frac{1}{N} \sum_{i=1}^{N} \|\boldsymbol{f}(\boldsymbol{x}_i) - \boldsymbol{y}_i\|^2 + \lambda \|\boldsymbol{f}\|_{\mathcal{F}}^2 \tag{2}$$

where the regularization parameter $\lambda > 0$ controls the smoothness of the learned function [16]. This approach may fail to generalize beyond the data set used to learn the dynamical model, which can cause the learned model to diverge significantly from the true system. Furthermore, if the trajectories in the data set are limited and noisy, the learned dynamical model may fail to capture the dynamics of the underlying system due to overfitting.

It is assumed that there is some information about the physical properties of the dynamical system. In particular, we

will study systems that are Hamiltonian, and where the function $\boldsymbol{f}(\boldsymbol{x})$ may be odd or even. This type of side information about the system was treated in [3] where the function class $\mathcal{F}$ was polynomial functions. The additional information on the dynamics was then included as side constraints by defining a subset $\mathcal{S}_i \in \mathcal{F}$ for each side constraint i, so that the function $\boldsymbol{f}^*$ satisfies the side constraint whenever $\boldsymbol{f}^* \in \mathcal{S}_i$. The minimization including the side constraints can then be formulated as the learning problem

$$\boldsymbol{f}^* = \arg\min_{\boldsymbol{f} \in \mathcal{F} \cap \mathcal{S}_1 \cap \cdots \cap \mathcal{S}_k} \frac{1}{N} \sum_{i=1}^{N} \|\boldsymbol{f}(\boldsymbol{x}_i) - \boldsymbol{y}_i\|^2 \tag{3}$$

In this paper the side constraints are instead handled by defining a reproducing kernel which ensures that the RKHS function class $\mathcal{F}$ inherently satisfies the relevant side constraints. It is well-known that this can be done to have an RKHS where the vector field $\boldsymbol{f}^*$ is curl-free, divergence-free [17], Hamiltonian [18], odd or even [6]. It is also possible to impose additional side constraints like contraction [2] or stabilizability [4] along the trajectories of the dataset, but this will not be addressed in this paper.

In this paper the function class $\mathcal{F}$ will be a reproducing kernel Hilbert space (RKHS). The side constraints are that the state dynamics are Hamiltonian and therefore symplectic, and, in addition, odd in the sense that $\boldsymbol{f}(-\boldsymbol{x}) = -\boldsymbol{f}(\boldsymbol{x})$, and this is ensured by selecting an appropriate reproducing kernel.

III. PRELIMINARIES

A. Reproducing kernel Hilbert space

This paper deals with the learning of functions in a reproducing kernel Hilbert space (RKHS) [14]. Real-valued scalar functions $f : \mathbb{R}^n \to \mathbb{R}$ are considered first. An RKHS $\mathcal{H}_k$ is then defined in terms of a function $k : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$. Let the function $k_x : \mathbb{R}^n \to \mathbb{R}$ be defined by $k_x(\boldsymbol{z}) = k(\boldsymbol{x}, \boldsymbol{z})$ for all $\boldsymbol{x}, \boldsymbol{z} \in \mathbb{R}^n$. The kernel function k is then a reproducing kernel and $\mathcal{H}_k$ is an RKHS if

$$\text{span}\{k_x : k_x(\boldsymbol{z}) = k(\boldsymbol{x}, \boldsymbol{z})\} \subseteq \mathcal{H}_k \tag{4}$$

and the reproducing property $f(\boldsymbol{x}) = \langle f, k_x \rangle_{\mathcal{H}_k}$ holds. It is noted that the reproducing property implies that

$$\langle k_x, k_z \rangle_{\mathcal{H}_k} = k_x(\boldsymbol{z}) = k(\boldsymbol{x}, \boldsymbol{z}) \tag{5}$$

According to the Moore-Aronszajn theorem [14] a function $k : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ is a reproducing kernel if and only if it is a positive definite kernel, which is the case whenever $k(\boldsymbol{x}, \boldsymbol{z}) = k(\boldsymbol{z}, \boldsymbol{x})$ for all $\boldsymbol{x}, \boldsymbol{z} \in \mathbb{R}^n$, and

$$\sum_{i=1}^{N} \sum_{j=1}^{N} c_i c_j k(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq 0 \tag{6}$$

for any set of points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \in \mathbb{R}^n$ and any set of real numbers $c_1, \ldots, c_N \in \mathbb{R}$. A function $f \in \mathcal{H}_k$ in the RKHS defined by the reproducing kernel k is given by

$$f(\boldsymbol{x}) = \sum_{i=1}^{N} a_i k(\boldsymbol{x}, \boldsymbol{x}_i) \in \mathbb{R} \tag{7}$$

The extension to vector-valued functions can be found in [15] and [19]. In this case the vector-valued function $\boldsymbol{f}: \mathbb{R}^n \to \mathbb{R}^n$ is to be learned. Let the matrix-valued kernel function $\boldsymbol{K}: \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^{n \times n}$ be a positive definite kernel, which means that

$$\boldsymbol{K}(\boldsymbol{x}, \boldsymbol{z}) = \boldsymbol{K}(\boldsymbol{z}, \boldsymbol{x})^{\mathrm{T}}, \quad \forall \, \boldsymbol{x}, \boldsymbol{z} \in \mathbb{R}^n \tag{8}$$

and, in addition,

$$\sum_{i=1}^{N} \sum_{j=1}^{N} \langle \boldsymbol{y}_i, \boldsymbol{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) \boldsymbol{y}_j \rangle \geq 0 \tag{9}$$

for every set of vectors $\{\boldsymbol{x}_i\}_{i=1}^{N} \in \mathbb{R}^n$ and $\{\boldsymbol{y}_i\}_{i=1}^{N} \in \mathbb{R}^n$. Then $\boldsymbol{K}$ is a reproducing kernel, and $\mathcal{H}_K$ is an RKHS. This corresponds to the Moore-Aronszajn theorem in the scalar case. Define the function $\boldsymbol{K}_x \boldsymbol{y}: \mathbb{R}^n \to \mathbb{R}^n$ as

$$(\boldsymbol{K}_x \boldsymbol{y})(\boldsymbol{z}) = \boldsymbol{K}(\boldsymbol{z}, \boldsymbol{x}) \boldsymbol{y} \in \mathbb{R}^n, \quad \forall \, \boldsymbol{z} \in \mathbb{R}^n \tag{10}$$

Then $\mathcal{H}_K$ is the closure of

$$\mathrm{span}\{\boldsymbol{K}_x \boldsymbol{y} \mid \boldsymbol{x} \in \mathbb{R}^n, \boldsymbol{y} \in \mathbb{R}^n\} \subseteq \mathcal{H}_K \tag{11}$$

and the reproducing property is given by

$$\langle \boldsymbol{f}(\boldsymbol{x}), \boldsymbol{y} \rangle = \langle \boldsymbol{f}, \boldsymbol{K}_x \boldsymbol{y} \rangle_{\mathcal{H}_K}, \quad \forall \, \boldsymbol{f} \in \mathcal{H}_K \tag{12}$$

The reproducing property implies that

$$\langle \boldsymbol{K}_z \boldsymbol{w}, \boldsymbol{K}_x \boldsymbol{y} \rangle_{\mathcal{H}_K} = \langle \boldsymbol{K}_z \boldsymbol{w}(\boldsymbol{x}), \boldsymbol{y} \rangle = \langle \boldsymbol{K}(\boldsymbol{x}, \boldsymbol{z}) \boldsymbol{w}, \boldsymbol{y} \rangle \tag{13}$$

Then functions in the RKHS $\mathcal{H}_K$ can be defined as

$$\boldsymbol{f} = \sum_{i=1}^{N} \boldsymbol{K}_{x_i} \boldsymbol{y}_i \in \mathcal{H}_K, \quad \boldsymbol{g} = \sum_{j=1}^{N} \boldsymbol{K}_{z_j} \boldsymbol{w}_j \in \mathcal{H}_K \tag{14}$$

with inner product

$$\langle \boldsymbol{f}, \boldsymbol{g} \rangle_{\mathcal{H}_K} = \sum_{i=1}^{N} \sum_{j=1}^{N} \langle \boldsymbol{y}_i, \boldsymbol{K}(\boldsymbol{x}_i, \boldsymbol{z}_j) \boldsymbol{w}_j \rangle_{\mathcal{H}_K} \tag{15}$$

and norm $\|\boldsymbol{f}\|_{\mathcal{H}_K}^2 = \langle \boldsymbol{f}, \boldsymbol{f} \rangle_{\mathcal{H}_K}$.

It is noted that (12) gives $\langle \boldsymbol{f}(\boldsymbol{x}), \boldsymbol{y} \rangle = \langle \boldsymbol{K}_x^* \boldsymbol{f}, \boldsymbol{y} \rangle$, where $\boldsymbol{K}_x^*$ is the adjoint of $\boldsymbol{K}_x$. It follows that $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{K}_x^* \boldsymbol{f}$, and

$$\|\boldsymbol{f}(\boldsymbol{x})\| = \|\boldsymbol{K}_x^* \boldsymbol{f}\| \leq \|\boldsymbol{K}_x^*\| \|\boldsymbol{f}\|_{\mathcal{H}_K} \leq \sqrt{\|\boldsymbol{K}(\boldsymbol{x}, \boldsymbol{x})\|} \|\boldsymbol{f}\|_{\mathcal{H}_K} \tag{16}$$

where $\sqrt{\|\boldsymbol{K}(\boldsymbol{x}, \boldsymbol{x})\|}$ is bounded [15]. This implies

$$\|\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{g}(\boldsymbol{x})\| \leq \sqrt{\|\boldsymbol{K}(\boldsymbol{x}, \boldsymbol{x})\|} \|\boldsymbol{f} - \boldsymbol{g}\|_{\mathcal{H}_K} \tag{17}$$

which shows that if $\|\boldsymbol{f} - \boldsymbol{g}\|_{\mathcal{H}_K}$ converges to zero, then $\|\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{g}(\boldsymbol{x})\|$ converges to zero for each $\boldsymbol{x}$.

B. Learning dynamical systems with RKHS

In this paper we aim to learn vector fields given by (1) where $\boldsymbol{y} = \dot{\boldsymbol{x}}$ is available. The estimation of the vector field is done using the vector-valued regularized least-squares problem [15]

$$\boldsymbol{f}^* = \underset{\boldsymbol{f} \in \mathcal{H}_K}{\arg\min} \frac{1}{N} \sum_{i=1}^{N} \|\boldsymbol{f}(\boldsymbol{x}_i) - \boldsymbol{y}_i\|^2 + \lambda \|\boldsymbol{f}\|_{\mathcal{H}_K}^2 \tag{18}$$

where $\mathcal{Z} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i) \in \mathbb{R}^n \times \mathbb{R}^n\}_{i=1}^{N}$ is the data used to learn the vector field, and $\lambda > 0$ is the regularization parameter. The function is given according to the representer theorem [20] as

$$\boldsymbol{f}^* = \sum_{i=1}^{N} \boldsymbol{K}(\cdot, \boldsymbol{x}_i) \boldsymbol{a}_i \in \mathcal{H}_K \tag{19}$$

where the optimal solution is given with the coefficients $\boldsymbol{a}_i \in \mathbb{R}^n$ found from [15]

$$\sum_{j=1}^{N} \boldsymbol{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) \boldsymbol{a}_j + N \lambda \boldsymbol{a}_j = \boldsymbol{y}_i \tag{20}$$

The function value of the optimal vector field is

$$\boldsymbol{f}^*(\boldsymbol{x}) = \sum_{i=1}^{N} \boldsymbol{K}(\boldsymbol{x}, \boldsymbol{x}_i) \boldsymbol{a}_i \in \mathbb{R}^n \tag{21}$$

A matrix formulation of (20) is found in [19].

C. Hamiltonian dynamics

Consider a holonomic system with generalized coordinates $\boldsymbol{q} = [q_1, \dots, q_m]^{\mathrm{T}}$ and Lagrangian [21]

$$L(\boldsymbol{q}, \dot{\boldsymbol{q}}, t) = T(\boldsymbol{q}, \dot{\boldsymbol{q}}, t) - U(\boldsymbol{q}) \tag{22}$$

The Lagrangian equation of motion is given by

$$\frac{d}{dt}\left(\frac{\partial L(\boldsymbol{q}, \dot{\boldsymbol{q}}, t)}{\partial \dot{\boldsymbol{q}}}\right) - \frac{\partial L(\boldsymbol{q}, \dot{\boldsymbol{q}}, t)}{\partial \boldsymbol{q}} = \boldsymbol{\tau} \tag{23}$$

The momentum vector $\boldsymbol{p} = [p_1, \dots, p_m]^{\mathrm{T}}$ is defined by

$$\boldsymbol{p} = \frac{\partial L(\boldsymbol{q}, \dot{\boldsymbol{q}}, t)}{\partial \dot{\boldsymbol{q}}}^{\mathrm{T}} \tag{24}$$

A change of coordinates from $(\boldsymbol{q}, \dot{\boldsymbol{q}})$ to $(\boldsymbol{q}, \boldsymbol{p})$ is introduced with the Legendre transformation [21]

$$H(\boldsymbol{q}, \boldsymbol{p}, t) = \boldsymbol{p}^{\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{q}, \boldsymbol{p}, t) - L(\boldsymbol{q}, \boldsymbol{\phi}(\boldsymbol{q}, \boldsymbol{p}, t), t) \tag{25}$$

where $\dot{\boldsymbol{q}} = \boldsymbol{\phi}(\boldsymbol{q}, \boldsymbol{p}, t)$. Hamilton's equations of motion are then given by

$$\dot{\boldsymbol{q}} = \frac{\partial H}{\partial \boldsymbol{p}}^{\mathrm{T}} \tag{26}$$

$$\dot{\boldsymbol{p}} = -\frac{\partial H}{\partial \boldsymbol{q}}^{\mathrm{T}} + \boldsymbol{\tau} \tag{27}$$

The time derivative of the Hamiltonian is seen to be

$$\frac{dH}{dt} = \frac{\partial H}{\partial \boldsymbol{p}} \dot{\boldsymbol{p}} + \frac{\partial H}{\partial \boldsymbol{q}} \dot{\boldsymbol{q}} + \frac{\partial H}{\partial t} = \dot{\boldsymbol{q}}^{\mathrm{T}} \boldsymbol{\tau} + \frac{\partial H}{\partial t} \tag{28}$$

If the Hamiltonian does not depend on time t and $\boldsymbol{\tau} = \boldsymbol{0}$, then

$$\frac{dH(\boldsymbol{q}, \boldsymbol{p})}{dt} = 0 \tag{29}$$

The numerical value of the Hamiltonian H will depend on the definition of the zero level of the potential $U(\boldsymbol{q})$.

D. Hamiltonian dynamics in the phase space

The phase space is defined as a $2m$-dimensional space with state vector $\boldsymbol{x} = \begin{bmatrix} \boldsymbol{q}^{\mathrm{T}}, \boldsymbol{p}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{2m}$. The Hamiltonian dynamics (26), (27) with $\boldsymbol{\tau} = \boldsymbol{0}$ can be given in the phase space as

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{J}\nabla H(\boldsymbol{x}) \tag{30}$$

where

$$\boldsymbol{J} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} \\ -\boldsymbol{I} & \boldsymbol{0} \end{bmatrix} \in \mathbb{R}^{2m \times 2m} \tag{31}$$

is the symplectic matrix. The flow in the phase space with initial condition $\boldsymbol{x}(0) = \boldsymbol{x}_0$ is given by

$$\phi_t(\boldsymbol{x}_0) = \boldsymbol{x}(t) \tag{32}$$

where $\boldsymbol{x}(t)$ is the solution of the Hamiltonian dynamics (30) at time t given the initial value $\boldsymbol{x}_0$.

E. Symplectic property of Hamiltonian dynamics

Consider any dynamic system

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) \tag{33}$$

where $\boldsymbol{x} \in \mathbb{R}^{2m}$. Define the Jacobian $\boldsymbol{\Psi}(t) = \partial\phi_t(\boldsymbol{x}_0)/\partial\boldsymbol{x}_0$. Then the system (33) is said to be symplectic if

$$\boldsymbol{\Psi}(t)^{\mathrm{T}}\boldsymbol{J}\boldsymbol{\Psi}(t) = \boldsymbol{J} \tag{34}$$

for all $t \geq 0$. The system (33) is Hamiltonian if and only if it is symplectic [22].

IV. METHOD

This section presents the main theoretical results of the paper.

A. Gaussian kernel

The Gaussian reproducing kernel is widely used in applications. In the scalar case it is defined as

$$k_\sigma(\boldsymbol{x}, \boldsymbol{z}) = e^{-\frac{\|\boldsymbol{x} - \boldsymbol{z}\|^2}{2\sigma^2}} \quad \in \mathbb{R} \tag{35}$$

where $\sigma > 0$. The Gaussian kernel is shift-invariant, since $k_\sigma(\boldsymbol{x}, \boldsymbol{z}) = g_\sigma(\boldsymbol{x} - \boldsymbol{z})$, where $g_\sigma(\boldsymbol{x}) = \exp-\frac{\|\boldsymbol{x}\|^2}{2\sigma^2}$. It is noted that the Gaussian kernel satisfies $k_\sigma(\boldsymbol{x}, \boldsymbol{z}) = k_\sigma(-\boldsymbol{x}, -\boldsymbol{z})$.

In the case of RKHS for vector-valued functions a separable Gaussian kernel can be used [2]. The separable Gaussian kernel is defined by

$$\boldsymbol{K}_\sigma(\boldsymbol{x}, \boldsymbol{z}) = k_\sigma(\boldsymbol{x}, \boldsymbol{z})\boldsymbol{I}_n \quad \in \mathbb{R}^n \tag{36}$$

B. Odd kernel

Consider a reproducing kernel which satisfies $k(\boldsymbol{x}, \boldsymbol{z}) = k(-\boldsymbol{x}, -\boldsymbol{z})$. Then

$$k_{\mathrm{odd}}(\boldsymbol{x}, \boldsymbol{z}) = \frac{1}{2}(k(\boldsymbol{x}, \boldsymbol{z}) - k(-\boldsymbol{x}, \boldsymbol{z})) \quad \in \mathbb{R} \tag{37}$$

is an odd reproducing kernel with an associated RKHS [6]. Any function $f(\boldsymbol{x}) = \sum_{i=1}^N a_i k_{\mathrm{odd}}(\boldsymbol{x}, \boldsymbol{x}_i)$ in the RKHS defined by k_{odd} will then be odd, since $k_{\mathrm{odd}}(-\boldsymbol{x}, \boldsymbol{z}) = -k_{\mathrm{odd}}(\boldsymbol{x}, \boldsymbol{z})$ and therefore $f(-\boldsymbol{x}) = -f(\boldsymbol{x})$.

C. Curl-free kernel

In the learning of a vector field $\boldsymbol{f}(\boldsymbol{x}) \in \mathbb{R}^n$ a curl-free reproducing kernel $\boldsymbol{K}_c(\boldsymbol{x}, \boldsymbol{z}) = \boldsymbol{G}_c(\boldsymbol{x} - \boldsymbol{z}) \in \mathbb{R}^{n \times n}$ can be derived from a shift-invariant reproducing kernel $k(\boldsymbol{x}, \boldsymbol{z}) = g(\boldsymbol{x} - \boldsymbol{z}) \in \mathbb{R}$, where k is typically the Gaussian kernel [23]. This is useful if it is required that the vector field is known to be curl-free.

The starting point for the derivation is the observation that the elements $\partial g(\boldsymbol{x})/\partial x_i$ for $i = 1, \ldots, n$ of the gradient vector

$$\nabla^{\mathrm{T}} g(\boldsymbol{x}) = \begin{bmatrix} \frac{\partial g(\boldsymbol{x})}{\partial x_1} & \cdots & \frac{\partial g(\boldsymbol{x})}{\partial x_n} \end{bmatrix} \tag{38}$$

are scalar functions which can be regarded as scalar fields. Then each column $-\nabla\frac{\partial g(\boldsymbol{x})}{\partial x_i}$ of the matrix

$$\boldsymbol{G}_c(\boldsymbol{x}) = -\nabla\nabla^{\mathrm{T}} g(\boldsymbol{x}) = \begin{bmatrix} -\nabla\frac{\partial g(\boldsymbol{x})}{\partial x_1} & \cdots & -\nabla\frac{\partial g(\boldsymbol{x})}{\partial x_n} \end{bmatrix} \tag{39}$$

is the gradient of a scalar field. The curl of the gradient of a scalar field is always zero, which implies that each column of $\boldsymbol{G}_c$ is curl-free. Any function in the RKHS of $\boldsymbol{G}_c$ will then be given by

$$\boldsymbol{f}(\boldsymbol{x}) = \sum_{i=1}^N \boldsymbol{G}_c(\boldsymbol{x} - \boldsymbol{x}_i)\boldsymbol{a}_i \tag{40}$$

where $\boldsymbol{f}(\boldsymbol{x})$ is a linear combination of the n columns of the N matrices $\boldsymbol{G}_c(\boldsymbol{x} - \boldsymbol{x}_i)$. Since each of these nN columns are curl-free it follows that the vector field $\boldsymbol{f}(\boldsymbol{x})$ is curl-free.

If g is selected as the Gaussian kernel, then the curl-free kernel is given by [23]

$$\boldsymbol{G}_c(\boldsymbol{x}) = -\nabla\nabla^{\mathrm{T}} g_\sigma(\boldsymbol{x}) = \frac{1}{\sigma^2}e^{-\frac{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}}{2\sigma^2}}\left(\boldsymbol{I} - \frac{\boldsymbol{x}\boldsymbol{x}^{\mathrm{T}}}{\sigma^2}\right) \tag{41}$$

which is the curl-free kernel used in [2].

D. Symplectic kernel

In [18] a symplectic kernel was presented for adaptive prediction of Hamiltonian dynamics. The symplectic kernel is based on the curl-free kernel in (39), and is given by

$$\boldsymbol{K}_s(\boldsymbol{x}, \boldsymbol{z}) = \boldsymbol{G}_s(\boldsymbol{x} - \boldsymbol{z}) = \boldsymbol{J}\boldsymbol{G}_c(\boldsymbol{x} - \boldsymbol{z})\boldsymbol{J}^{\mathrm{T}} \tag{42}$$

To verify that functions in the resulting RKHS are symplectic it is used that a function in the RKHS of $\boldsymbol{G}_s$ is given by

$$\boldsymbol{f}(\boldsymbol{x}) = \sum_{i=1}^N \boldsymbol{G}_s(\boldsymbol{x} - \boldsymbol{x}_i)\boldsymbol{a}_i \tag{43}$$

which gives

$$\boldsymbol{f}(\boldsymbol{x}) = -\sum_{i=1}^N \boldsymbol{J}\nabla\nabla^{\mathrm{T}} g(\boldsymbol{x} - \boldsymbol{x}_i)\boldsymbol{J}^{\mathrm{T}}\boldsymbol{a}_i \tag{44}$$

$$= -\boldsymbol{J}\nabla\sum_{i=1}^N \nabla^{\mathrm{T}} g(\boldsymbol{x} - \boldsymbol{x}_i)\boldsymbol{c}_i \tag{45}$$

where $\boldsymbol{c}_i = \boldsymbol{J}^{\mathrm{T}}\boldsymbol{a}_i$. This results in the Hamiltonian dynamics

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{J}\nabla H(\boldsymbol{x}) \tag{46}$$

where the Hamiltonian is

$$H(\boldsymbol{x}) = -\sum_{i=1}^{N} \nabla^{\mathrm{T}} g(\boldsymbol{x} - \boldsymbol{x}_i) \boldsymbol{c}_i \tag{47}$$

In [18], the function g was the Gaussian kernel (35). Then the curl-free kernel is given by (41), and the resulting symplectic kernel is

$$\boldsymbol{G}_s(\boldsymbol{x}) = \frac{1}{\sigma^2} e^{-\frac{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}}{2\sigma^2}} \boldsymbol{J} \left(\boldsymbol{I} - \frac{\boldsymbol{x}\boldsymbol{x}^{\mathrm{T}}}{\sigma^2} \right) \boldsymbol{J}^{\mathrm{T}} \tag{48}$$

E. Odd symplectic kernel

Next, we propose an odd symplectic kernel to ensure that the resulting dynamics are Hamiltonian and odd symmetric. The starting point is an odd Gaussian kernel

$$k_{\sigma,\mathrm{odd}}(\boldsymbol{x}, \boldsymbol{z}) = \frac{1}{2} \left(e^{-\frac{\|\boldsymbol{x}-\boldsymbol{z}\|^2}{2\sigma^2}} - e^{-\frac{\|-\boldsymbol{x}-\boldsymbol{z}\|^2}{2\sigma^2}} \right) \tag{49}$$

$$= \frac{1}{2} \left(e^{-\frac{\|\boldsymbol{x}-\boldsymbol{z}\|^2}{2\sigma^2}} - e^{-\frac{\|\boldsymbol{x}+\boldsymbol{z}\|^2}{2\sigma^2}} \right) \tag{50}$$

The odd curl-free kernel is then derived from

$$\boldsymbol{K}_{c,\mathrm{odd}}(\boldsymbol{x}, \boldsymbol{z}) = -\nabla\nabla^{\mathrm{T}} k_{\sigma,\mathrm{odd}}(\boldsymbol{x}, \boldsymbol{z}) \tag{51}$$

The first partial derivative is

$$\frac{\partial k_{\sigma,\mathrm{odd}}}{\partial x_i} = -\frac{1}{2\sigma^2} \left((x_i - z_i) e^{-\frac{\|\boldsymbol{x}-\boldsymbol{z}\|^2}{2\sigma^2}} - (x_i + z_i) e^{-\frac{\|\boldsymbol{x}+\boldsymbol{z}\|^2}{2\sigma^2}} \right)$$

The second partial derivative is

$$\frac{\partial^2 k_{\sigma,\mathrm{odd}}}{\partial x_i \partial x_j} =$$
$$-\frac{1}{2\sigma^2} \left(e^{-\frac{\|\boldsymbol{x}-\boldsymbol{z}\|^2}{2\sigma^2}} \delta_{i,j} - e^{-\frac{\|\boldsymbol{x}-\boldsymbol{z}\|^2}{2\sigma^2}} \frac{(x_i - z_i)(x_j - z_j)}{\sigma^2} \right.$$
$$\left. - e^{-\frac{\|\boldsymbol{x}+\boldsymbol{z}\|^2}{2\sigma^2}} \delta_{i,j} + e^{-\frac{\|\boldsymbol{x}+\boldsymbol{z}\|^2}{2\sigma^2}} \frac{(x_i + z_i)(x_j + z_j)}{\sigma^2} \right) \tag{52}$$

where $\delta_{i,j} = 1$ when $i = j$ and 0 otherwise. Finally, the expression for the odd symmetric curl-free kernel is written as

$$\boldsymbol{K}_{c,\mathrm{odd}}(\boldsymbol{x}, \boldsymbol{z}) = \frac{1}{2\sigma^2} \left(e^{-\frac{\|\boldsymbol{x}-\boldsymbol{z}\|^2}{2\sigma^2}} \left(\boldsymbol{I} - \frac{(\boldsymbol{x}-\boldsymbol{z})^{\mathrm{T}}(\boldsymbol{x}-\boldsymbol{z})}{\sigma^2} \right) \right.$$
$$\left. - e^{-\frac{\|\boldsymbol{x}+\boldsymbol{z}\|^2}{2\sigma^2}} \left(\boldsymbol{I} - \frac{(\boldsymbol{x}+\boldsymbol{z})^{\mathrm{T}}(\boldsymbol{x}+\boldsymbol{z})}{\sigma^2} \right) \right) \tag{53}$$

This could also be found from (41) as

$$\boldsymbol{K}_{c,\mathrm{odd}}(\boldsymbol{x}, \boldsymbol{z}) = \frac{1}{2} \left(\boldsymbol{K}_c(\boldsymbol{x}, \boldsymbol{z}) - \boldsymbol{K}_c(-\boldsymbol{x}, \boldsymbol{z}) \right) \tag{54}$$

Finally, the odd symplectic kernel is written by combining (42) and (54), which gives

$$\boldsymbol{K}_{s,\mathrm{odd}}(\boldsymbol{x},\boldsymbol{z}) = \boldsymbol{J} \boldsymbol{K}_{c,\mathrm{odd}}(\boldsymbol{x}, \boldsymbol{z}) \boldsymbol{J}^{\mathrm{T}} \tag{55}$$

$$= \frac{1}{2} \boldsymbol{J} \left(\boldsymbol{K}_c(\boldsymbol{x}, \boldsymbol{z}) - \boldsymbol{K}_c(-\boldsymbol{x}, \boldsymbol{z}) \right) \boldsymbol{J}^{\mathrm{T}} \tag{56}$$

$$= \frac{1}{2} \left(\boldsymbol{J}\boldsymbol{K}_c(\boldsymbol{x}, \boldsymbol{z})\boldsymbol{J}^{\mathrm{T}} - \boldsymbol{J}\boldsymbol{K}_c(-\boldsymbol{x}, \boldsymbol{z})\boldsymbol{J}^{\mathrm{T}} \right) \tag{57}$$

$$= \frac{1}{2} \left(\boldsymbol{K}_s(\boldsymbol{x}, \boldsymbol{z}) - \boldsymbol{K}_s(-\boldsymbol{x}, \boldsymbol{z}) \right) \tag{58}$$

where $\boldsymbol{K}_s$ is defined as in (48).

V. EXPERIMENTS

In this section, the results of two numerical experiments are presented. The Hamiltonian dynamics were learned for two different Hamiltonian systems with odd vector fields. The regularized least-squares problem in (18) was solved using (20), and the resulting learned vector field was given by (21). The Gaussian separable kernel in (36) and the odd symplectic kernel in (58) were compared for the two systems.

A. Hyperparameter tuning

The hyperparameters σ and λ were tuned for each model by minimizing the cross-validation error [24] over the data set $\mathcal{Z} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i) \in \mathbb{R}^n \times \mathbb{R}^n\}_{i=1}^N$. The data set $\mathcal{Z}$ was split into mutually exclusive subsets $\mathcal{Z}_1, \ldots \mathcal{Z}_k$, and for each iteration $i \in \{1, \ldots, k\}$, the model was trained on the subset $\hat{\mathcal{Z}}_i = \mathcal{Z} \setminus \mathcal{Z}_i$ and evaluated on $\mathcal{Z}_i$. Formally, the hyperparameter optimization is written as [6]

$$\min_{\sigma, \lambda} \frac{1}{k} \sum_{i=1}^{k} \mathrm{MSE}\left(\boldsymbol{f}_{\hat{\mathcal{Z}}_i}, \mathcal{Z}_i \right) \tag{59}$$

where $\boldsymbol{f}_{\hat{\mathcal{Z}}_i}$ is the learned functions trained on the training set $\hat{\mathcal{Z}}_i = \mathcal{Z} \setminus \mathcal{Z}_i$, using the hyperparameters σ and λ, $\mathcal{Z}_i$ is the test set, and MSE is the empirical mean square error between the learned model and the test set. The optimization problem was solved using a grid search.

B. Harmonic oscillator

A harmonic oscillator is given as the undamped mass-spring system

$$\dot{x} = v, \qquad \dot{v} = -\frac{k}{m}x \tag{60}$$

where m is the mass, k is the spring constant, x is the position of the mass, and v is the velocity of the mass. The kinetic energy is $T = \frac{1}{2}m\dot{x}^2$ and the potential energy is $U = \frac{1}{2}kx^2$. The generalized coordinate is selected as $q = x$ and the momentum is then $p = m\dot{x}$. The state vector is $\boldsymbol{x} = [q, p]^{\mathrm{T}}$. The Hamiltonian for the system is

$$H(q, p) = T(p) + U(q) = \frac{1}{2}\frac{p^2}{m} + \frac{1}{2}kq^2 \tag{61}$$

The Hamiltonian dynamics of the true system are then given by

$$\dot{q} = \frac{\partial H}{\partial p} = \frac{p}{m}, \qquad \dot{p} = -\frac{\partial H}{\partial q} = -kq \tag{62}$$

Figure 1a shows phase curves of the true system as described with parameters $m = 0.5$ and $k = 1$. Figure 1a also shows three trajectories that were generated by simulation of the true system with three different initial conditions: $\boldsymbol{x}_{1,0} = [1, 0]^{\mathrm{T}}$, $\boldsymbol{x}_{2,0} = [2.25, 0]^{\mathrm{T}}$, and $\boldsymbol{x}_{3,0} = [3.5, 0]^{\mathrm{T}}$. The time step was $h = 0.25\,\mathrm{s}$ and the system was simulated for $t \in [0, 1]$ seconds, which resulted in 5 data points for each trajectory, and $N = 15$ total data points. The velocities $\boldsymbol{y}$ were sampled at each trajectory point, and zero mean Gaussian noise with standard deviation $\sigma_n = 0.1$ was added to the trajectory and velocity data. Figure 1b shows the resulting data set.

(a) True system   (b) Data set   (c) Learned gaussian model   (d) Learned odd sympl. model
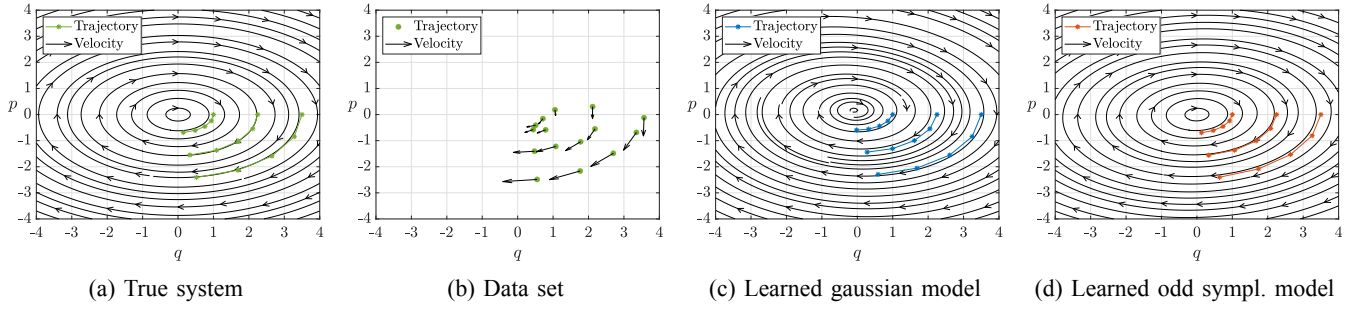
Fig. 1: Stream and trajectory plots for the harmonic oscillator and extracted data set, and the resulting learned models using the separable Gaussian kernel and the odd symplectic kernel.
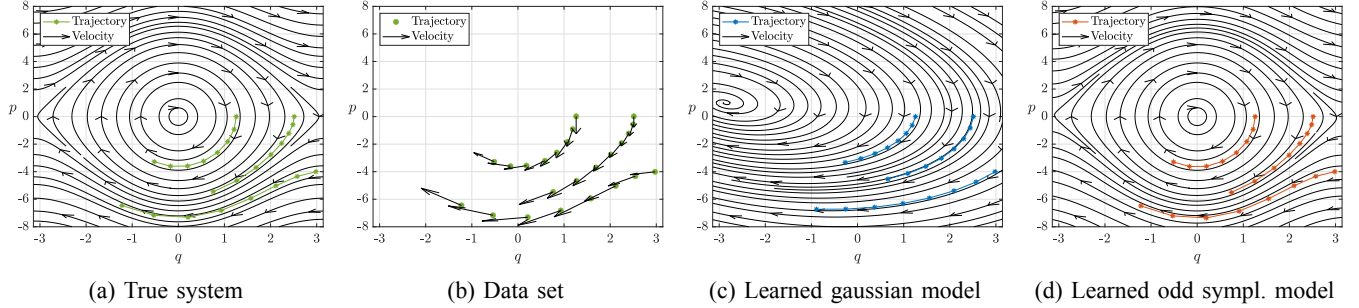

(a) True system   (b) Data set   (c) Learned gaussian model   (d) Learned odd sympl. model

Fig. 2: Stream and trajectory plots for the simple pendulum and extracted data set, and the resulting learned models using the separable Gaussian kernel and the odd symplectic kernel.

Cross-validation was used for tuning of the hyperparameters, and gave a kernel width $\sigma_g = 19.5$ and regularization parameter $\lambda_g = 10^{-4}$ for the Gaussian separable model, and kernel width $\sigma_{\text{os}} = 12.1$ and regularization parameter $\lambda_{\text{os}} = 10^{-4}$ for the odd symplectic model.

Figures 1c and 1d show the learned models when the separable Gaussian kernel and the odd symplectic kernel were used. Both models generalized well, but the odd symplectic model recreated the periodic orbits of the true system shown in Figure 1a, indicating that the energy conservation was captured in the model. The lack of periodic orbits in Figure 1c documents a lack of energy conservation in the separable Gaussian model.

A separate test trajectory was simulated to test how well the learned models generalize for unknown data. The initial condition was $\boldsymbol{x}_0 = \begin{bmatrix} 2, 0 \end{bmatrix}^{\text{T}}$ and the time horizon was $t \in \begin{bmatrix} 0, 4 \end{bmatrix}$ seconds. The error between the true system and the learned model trajectories is given by Err $= \|\boldsymbol{x}_b - \boldsymbol{x}_l\|$. Figure 3a shows the three resulting trajectories, and Figure 3b shows the error for each time step. The results show that the odd symplectic model tracked the true system trajectory with an order of magnitude improvement in accuracy, which indicates the improvement in generalization with the odd, symplectic kernel.

C. Simple pendulum

A simple pendulum is modeled with a point mass m at the end of a mass-less rod of length l. The pendulum angle is θ. The equation of motion is given by

$$\ddot{\theta} = -\frac{g}{l}\sin(\theta) \qquad (63)$$

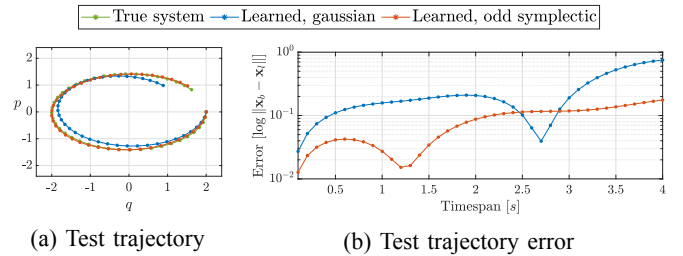(a) Test trajectory   (b) Test trajectory error

Fig. 3: Comparison of the two learned models against the harmonic oscillator system, using the test trajectory.

where g is the acceleration of gravity. The generalized coordinate is $q = \theta$, the kinetic energy is $T = \frac{1}{2}ml^2\dot{q}^2$ and the potential energy is $U = mgl(1 - \cos(q))$. The Lagrangian is defined as $L = T - U$ and from (24) the generalized momentum is

$$p = \frac{\partial L}{\partial \dot{q}} = ml^2\dot{q} \qquad (64)$$

The Hamiltonian is

$$H(q,p) = p\dot{q} - L = \frac{p^2}{2ml^2} + mgl(1 - \cos(q)) \qquad (65)$$

The Hamiltonian dynamics are then given by

$$\dot{q} = \frac{\partial H}{\partial p} = \frac{p}{ml^2}, \qquad \dot{p} = -\frac{\partial H}{\partial q} = -mgl\sin(q) \qquad (66)$$

Figure 2a shows the true system with parameters $m = 0.5$, $l = 1$, and $g = 9.81$. Three trajectories were generated, and the system was simulated with three different initial conditions: $\boldsymbol{x}_{1,0} = \begin{bmatrix} \frac{2\pi}{5}, 0 \end{bmatrix}^{\text{T}}$, $\boldsymbol{x}_{2,0} = \begin{bmatrix} \frac{4\pi}{5}, 0 \end{bmatrix}^{\text{T}}$, and $\boldsymbol{x}_{3,0} = \begin{bmatrix} \frac{19\pi}{20}, -4 \end{bmatrix}^{\text{T}}$. The time step was set to $h = 0.1\,\text{s}$ and the system was simulated for $t \in \begin{bmatrix} 0, 0.7 \end{bmatrix}$ seconds, giving 8

data points for each trajectory, and $N = 24$ total data points. The velocities $\boldsymbol{y}$ were sampled at each trajectory point, and zero mean Gaussian noise with standard deviation $\sigma_n = 0.01$ was added to the trajectory and velocity data. Figure 2b shows the resulting data set.

The cross-validation procedure gave kernel width $\sigma_g = 12.3$ and regularization parameter $\lambda_g = 0.1$ for the Gaussian separable model, and kernel width $\sigma_{\mathrm{os}} = 3$ and regularization parameter $\lambda_{\mathrm{os}} = 10^{-4}$ for the odd symplectic model.

Figures 2c and 2d show the learned models using the separable Gaussian kernel and the odd symplectic kernel, respectively. The function learned with the Gaussian separable kernel did not give an accurate representation of the true dynamics from such a limited dataset. The model learned with the odd symplectic kernel was accurate and gave a good representation of the vector field of the simple pendulum system. It is seen from Figure 2d that symmetry and energy conservation lead to periodic orbits.

A separate test trajectory was simulated to test the generalized performance of the learned models. The initial condition was $\boldsymbol{x}_0 = \begin{bmatrix} \frac{\pi}{2}, 0 \end{bmatrix}^{\mathrm{T}}$ and the time horizon was $t \in \begin{bmatrix} 0, 2 \end{bmatrix}$ seconds. The error between the true system and the learned model trajectories was defined as $\mathrm{Err} = \|\boldsymbol{x}_b - \boldsymbol{x}_l\|$. Figure 4a shows the three resulting trajectories, and Figure 4b shows the error for each time step. The results show that the odd symplectic model is far more accurate than the Gaussian separable model, which fails to generalize beyond the area close to the data set.
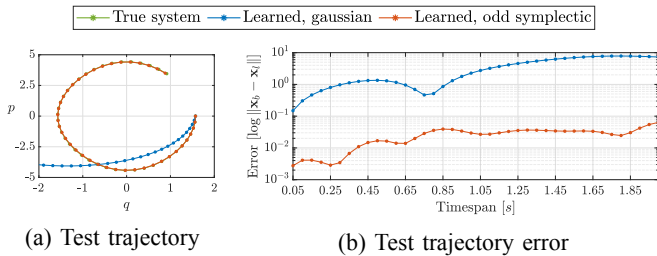

(a) Test trajectory   (b) Test trajectory error

Fig. 4: Comparison of the two learned models against the simple pendulum system, using the test trajectory.

D. Numerical evaluation

The models were evaluated numerically to investigate the ability of the learned models to capture the side information of the true systems. The odd symmetry was evaluated by sampling $10\,000$ points in the right half plane of the phase portraits shown in Figure 1 and Figure 2, and calculating the odd error given as

$$e_{\mathrm{odd}} = \|\boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{f}(-\boldsymbol{x})\| \tag{67}$$

where $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^n$ is the dynamical system being evaluated and $\boldsymbol{x} \in \mathbb{R}^n$ is the sampled point.

The results in Table I document that the learned odd symplectic model enforces odd symmetry like the true systems, whereas the Gaussian separable model does not. The learned Hamiltonian in (47) for the learned odd symplectic model

TABLE I: Odd error e_{odd} for the two dynamical systems

System	Harmonic oscillator		Simple pendulum	
	Mean	Variance	Mean	Variance
True e_{odd}	0.00	0.00	0.00	0.00
Gaussian sep. e_{odd}	0.65	0.08	7.87	1.49
Odd symplectic e_{odd}	0.00	0.00	0.00	0.00

was evaluated over the test trajectories shown in Figure 3 and Figure 4, and compared to their corresponding real Hamiltonians in (61) and (65).

The results in Table II demonstrate that the value of the learned Hamiltonian $\hat{H}(\boldsymbol{x})$ has a constant offset from the true Hamiltonian $H(\boldsymbol{x})$. This agrees with the fact that the potential energy's zero potential cannot be expected to be the same for the learned and true systems. It is seen that the value of the learned Hamiltonian is constant in agreement with (29) since the system is unforced and independent of time. This is reflected in the variance of both $H(\boldsymbol{x})$ and $\hat{H}(\boldsymbol{x})$. Noting that these are results from numerical simulations, the results indicate that the Hamiltonian mechanics are captured in the learned odd symplectic model.

TABLE II: Hamiltonian for the two dynamical systems

Hamiltonian	Harmonic oscillator		Simple pendulum	
	Mean	Variance	Mean	Variance
Real $H(\boldsymbol{x})$	1.99	$5.43 \cdot 10^{-9}$	9.81	$2.2 \cdot 10^{-6}$
Learned $\hat{H}(\boldsymbol{x})$	-108.54	$6.24 \cdot 10^{-9}$	-16.85	$1.34 \cdot 10^{-6}$

VI. CONCLUSION

The learning of Hamiltonian mechanical systems with odd vector fields using a specialized kernel has been investigated. The proposed method enforces side information concerning Hamiltonian dynamics and odd vector fields using the kernel rather than through constraints in an optimization problem. The encoding of side information in the kernel allows for a closed-form solution to the learning problem and enforces the constraints over the whole domain of the learned function. Comparative experiments on limited data sets demonstrate how the proposed kernel learns the dynamics of a system compared to a kernel that does not enforce side information. The standard Gaussian kernel recreates the system's dynamics close to the data set, but fails to capture the general characteristics of the system. The proposed kernel enforces side information and produces a model that generalizes well, capturing the dynamics even on a limited and noisy data set.

Future work: The proposed method could be applied to more complex mechanical systems. For problems with larger data sets, approximating the kernel using random Fourier features [25] could reduce the problem size, reducing the training and inference time of the learned model. Generalized momenta and their derivatives might not be available in practice, so extending the method to learn Hamiltonian systems from generalized coordinates and velocities only should be investigated. Finally, control-oriented learning could be studied using the proposed kernel.

REFERENCES

[1] S. L. Brunton and J. N. Kutz, *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*, 2nd ed. Cambridge University Press, 2022.

[2] V. Sindhwani, S. Tu, and M. Khansari, "Learning Contracting Vector Fields For Stable Imitation Learning," 2018, arXiv preprint arXiv:1804.04878 [cs.RO].

[3] A. A. Ahmadi and B. E. Khadir, "Learning Dynamical Systems with Side Information," in *Proceedings of the 2nd Conference on Learning for Dynamics and Control*, ser. Proceedings of Machine Learning Research, vol. 120. PMLR, 2020, pp. 718–727.

[4] S. Singh, S. M. Richards, V. Sindhwani, J.-J. E. Slotine, and M. Pavone, "Learning stabilizable nonlinear dynamics with contraction-based regularization," *The International Journal of Robotics Research*, vol. 40, no. 10-11, pp. 1123–1150, 2021.

[5] M. Revay and I. Manchester, "Contracting Implicit Recurrent Neural Networks: Stable Models with Improved Trainability," in *Proceedings of the 2nd Conference on Learning for Dynamics and Control*, ser. Proceedings of Machine Learning Research, vol. 120. PMLR, 2020, pp. 393–403.

[6] M. Krejnik and A. Tyutin, "Reproducing Kernel Hilbert Spaces With Odd Kernels in Price Prediction," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 10, pp. 1564–1573, 2012.

[7] C.-A. Cheng and H.-P. Huang, "Learn the Lagrangian: A Vector-Valued RKHS Approach to Identifying Lagrangian Systems," *IEEE Transactions on Cybernetics*, vol. 46, no. 12, pp. 3247–3258, 2016.

[8] M. Ahmadi, U. Topcu, and C. Rowley, "Control-Oriented Learning of Lagrangian and Hamiltonian Systems," in *American Control Conference (ACC)*, 2018, pp. 520–525.

[9] S. Greydanus, M. Dzamba, and J. Yosinski, "Hamiltonian Neural Networks," in *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

[10] Y. D. Zhong, B. Dey, and A. Chakraborty, "Symplectic ODE-Net: Learning Hamiltonian Dynamics with Control," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020.

[11] Z. Chen, J. Zhang, M. Arjovsky, and L. Bottou, "Symplectic Recurrent Neural Networks," in *International Conference on Learning Representations*, 2020.

[12] M. Khosravi and R. S. Smith, "Nonlinear System Identification With Prior Knowledge on the Region of Attraction," *IEEE Control Systems Letters*, vol. 5, no. 3, pp. 1091–1096, 2021.

[13] A. J. Thorpe, C. Neary, F. Djeumou, M. M. K. Oishi, and U. Topcu, "Physics-Informed Kernel Embeddings: Integrating Prior System Knowledge with Data-Driven Control," 2023, arXiv preprint arXiv:2301.03565 [eess.SY].

[14] N. Aronszajn, "Theory of Reproducing Kernels," *Transactions of the American Mathematical Society*, vol. 68, no. 3, pp. 337–404, 1950.

[15] C. A. Micchelli and M. Pontil, "On Learning Vector-Valued Functions," *Neural Computation*, vol. 17, no. 1, pp. 177–204, 2005.

[16] G. Pillonetto, T. Chen, A. Chiuso, G. D. Nicolao, and L. Ljung, *Regularized System Identification*, 1st ed. Springer Cham, 2022.

[17] M. Micheli and J. A. Glaunès, "Matrix-valued kernels for shape deformation analysis," *Geometry, Imaging and Computing*, vol. 1, no. 1, pp. 57—-139, 2014.

[18] N. M. Boffi, S. Tu, and J.-J. E. Slotine, "Nonparametric adaptive control and prediction: theory and randomized algorithms," *Journal of Machine Learning Research*, vol. 23, no. 281, pp. 1–46, 2022.

[19] H. Q. Minh and V. Sindhwani, "Vector-valued manifold regularization," in *Proceedings of the 28'th International Conference on Machine Learning*, 2011.

[20] B. Schölkopf, R. Herbrich, and A. J. Smola, "A Generalized Representer Theorem," in *Computational Learning Theory*. Springer Berlin Heidelberg, 2001, pp. 416–426.

[21] H. Goldstein, C. Poole, and J. Safko, *Classical Mechanics*, 3rd ed. Addison Wesley, 2002.

[22] E. Hairer, G. Wanner, and C. Lubich, *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations*, 2nd ed. Springer Berlin, Heidelberg, 2006.

[23] E. J. Fuselier, "Refined Error Estimates for Matrix-Valued Radial Basis Functions," 2006, PhD Thesis, Texas A&M University.

[24] R. Kohavi, "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection," in *Proceedings of the 14th International Joint Conference on Artificial Intelligence, IJCAI-95*, 1995, p. 1137–1143.

[25] A. Rahimi and B. Recht, "Random Features for Large-Scale Kernel Machines," in *Advances in Neural Information Processing Systems*, vol. 20, 2007.