Data-Induced Intelligent Kalman Filtering for Beam Prediction and Tracking of Millimeter Wave Communications

Jianjun Zhang*†, Yongming Huang†‡, Christos Masouros§, and Xiaohu You†‡
* College of Computer Science and Technology
Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China
† The Purple Mountain Laboratories, Nanjing 211111, China.
‡ National Mobile Communications Research Laboratory, Southeast University, Nanjing 210096, China
§ Department of Electronic & Electrical Engineering, University College London, London WC1E7JE, U.K.
Email: jianjun.zhang@nuaa.edu.cn, {huangym, xhyu}@seu.edu.cn, c.masouros@ucl.ac.uk.

***Abstract*—Beam prediction and tracking (BPT) are key technology for millimeter wave communications. Typical techniques include Kalman filtering (KF) and Gaussian process (GP) regression. However, KF requires explicit system dynamics, which is difficult to obtain for complicated scenarios. In contrast, thanks to the data-driven manner, GP regression circumvents this challenging, which, however, suffers from prohibitive computational complexity. To tackle this issue, we propose a novel hybrid model and data driven approach, referred to as data-induced intelligent Kalman filtering (DIIKF). DIIKF learns the system dynamics via the data-driven manner, which can enjoy the advantages of both KF and GP while overcoming their drawbacks. In view that the system dynamics is available, we further propose long-term prediction and design an efficient algorithm. Simulation results show that our method approaches the optimal oracle solution (in terms of effective achievable rate), with the linear complexity order.**

***Index Terms*—Beam prediction, beam tracking, hybrid model and data driven, Gaussian process, intelligent Kalman filtering.**

I. Introduction

Millimeter wave (mmwave) communications, occupying 30-300 GHz spectrum resources and offering significant underutilized bandwidth, have been considered as one of the most promising solutions to meet high-speed wireless data demands in the era of 5G and beyond [1]. To estimate the channel state information for mmwave communications, different methodologies have been investigated, among which the most widely accepted one is beam training and tracking [2]–[4].

In general, the beam training and tracking scheme consists of two stages, i.e., initial beam alignment and beam tracking. In the first stage, the optimal beam is often found via hierarchical or adaptive search [2]–[4]. As a result, large overhead, caused by wide-range beam sweeping, is involved in this stage. To alleviate this issue, beam tracking is invoked in the second stage, so as to avoid frequent search. Compared to the initial beam alignment, the number of beams used for tracking should be as small as possible. If the tracking fails, the search-based alignment operation is invoked again. It is hoped that both the frequency of alignment and the number of beams used for tracking can be small. Unfortunately, they are conflicting goals.

To mitigate the conflict, the vital means is beam prediction. Apparently, it is expected that the beam subspace predicted can be as small as possible, while still containing the real beam. This design goal constitutes the most important performance metric, which is referred to as prediction efficiency or success rate. As a key operation close to the physical environments, the complexity of beam prediction (including sample complexity and inference complexity) greatly determines the usability of an algorithm in practice. Hence, a beam prediction algorithm having good real-time performance is desired. Till now, various algorithms have been proposed, which roughly fall into two categories, i.e., the classical KF based methods [5]–[7] and the recent machine learning (ML) based methods [8]–[13].

For almost any prediction algorithm, the most crucial step is to construct an appropriate prediction model. The KF technique addresses this issue by explicitly building a dynamical model for the underlying physical system. Specifically, two stochastic differential equations (SDEs), referred to as state-space and measurement equations, are established. Then, the Kalman filter is invoked to recursively predict future states. A prominent advantage of the KF method is low computational complexity, which enables real-time applications. In particular, the scaling of the computational complexity is linear in terms of the number of sample points $\mathcal{O}(N)$, as opposed to the cubic scaling $\mathcal{O}(N^3)$ of GP regression. Unfortunately, the dynamics model is almost always derived manually, which thus limits its application scope, especially for complicated environments.

In contrast to KF, ML addresses the issue of prediction modeling by employing the data-driven mode. In fact, the salient ability of ML is that it can automatically extract meaningful patterns and derive an appropriate model from observed samples directly. The ML-based beam prediction methods fall into two categories, i.e., reinforcement learning based algorithms [11]–[15] and supervised learning based algorithms (including GP-based algorithms) [16]–[18]. However, the existing ML-based algorithms suffer from low convergence rate (e.g., the reinforcement learning based solutions), large sample complexity (e.g., supervised learning based methods) or large computational complexity (e.g., the GP-based designs). Moreover, most of them can only predict the beam for the most recent time-slot, which still requires frequent beam sweeping.

To tackle the aforementioned issues, in this paper we propose a novel hybrid model and data driven approach, referred to as DIIKF. DIIKF can exploit the advantages of both KF and ML methods while overcoming their drawbacks. In contrast to the GP-based algorithms, whose scaling of the computational complexity is cubic, the scaling of DIIKF is linear. Moreover, inheriting from KF, DIIKF is also applicable to non-stationary scenarios. Compared to the conventional KF-based methods, there is no need to manually derive the state and measurement equations, because they can be implicitly learned via the data-driven manner. These features enable real-time applications in practice. In view that the system dynamics has been available, we propose to predict the long-term behavior of the underlying beam process, which can lower the frequency of beam training and thus further improves system performance.

II. System Model

Consider a mmwave communication system, which consists of one base station (BS) equipped with N transmit antennas and K single-antenna users. Typical scenarios include indoor or outdoor communications (e.g., a pedestrian walks along a street), vehicle-to-everything (V2X), and so on. Without loss of generality, we take the V2X scenario as an example. To facilitate system implementation, we consider the codebook-based analog beamforming. The beams are chosen from a predefined codebook $\mathcal{C}$ of size M, i.e., $\mathcal{C} = \{\mathbf{f}_1, \cdots, \mathbf{f}_M\}$.

Due to the sparsity of mmwave channels, an extended Saleh-Valenzuela geometric model is considered here. The channel vector between the BS and user k is given by

$$\mathbf{h}_k = \sqrt{N/\beta} \sum_{l=1}^{L_k} \alpha_l \mathbf{a}(\phi_l, \psi_l), \tag{1}$$

where β is the average path-loss, L_k is the number of paths, and α_l is the complex path gain of the l-th path. In Eq.(1), ϕ_l and ψ_l represent the elevation angle and azimuth angle of the l-th path, respectively. For simplicity, the case of single-user and uniform linear array is investigated in this paper, and the subscript k is omitted next.

The signal received by the user for beam $\mathbf{f}_i \in \mathcal{C}$ is given by

$$y_i = \sqrt{P}\mathbf{h}^{\mathrm{H}}\mathbf{f}_i s + w_i, \tag{2}$$

where P is the transmit power, s with $|s| = 1$ is pilot symbol, and $w_i \sim \mathcal{CN}(0, 1)$ is received noise. The effective achievable rate is often used to measure the throughput performance [19]

$$R_{\text{eff}} = (1 - T_{\text{B}}/T_{\text{S}}) \log\left(1 + P|\mathbf{h}^{\mathrm{H}}\mathbf{f}_i|^2\right), \tag{3}$$

where T_{B} and T_{S} denote the duration of beam training within a time-slot and the duration of entire time-slot, respectively.

It can be observed from (3) that to achieve a high throughput, the time allocated for beam training T_{B} should be as little as possible, so as to reserve more time for data transmission. To reduce T_{B}, the key is to design a good prediction algorithm, which enables to predict a small but correct beam subspace. Next, we propose DIIKF to realize this nontrivial goal.

III. Data-Induced Intelligent Kalman Filtering

Motivated by the fact that KF is inference-efficient (via the Kalman filter) but modeling-inefficient and GP is modeling-efficient but inference-inefficient, we propose DIIKF to exploit the advantages of KF/GP while overcoming their drawbacks.

A. Outline and Theoretical Foundation

The core of DIIKF is to model and learn system dynamics via the ML techniques (e.g., the GP in this paper) and execute statistical inference (e.g., prediction) via KF (e.g., the discrete-time Kalman filter), as shown in Fig. 1. The key components of DIIKF include data collection, implicit representation of system dynamics (via GP), explicit representation of system dynamics (via SDE), and model representation conversion.

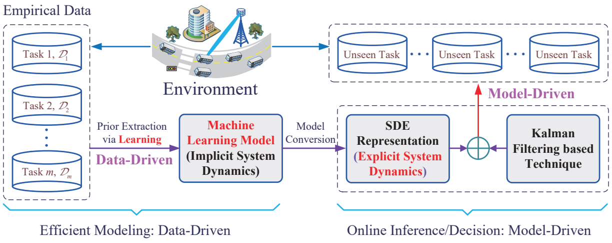

Empirical Data
Task 1, $\mathcal{D}_1$
Task 2, $\mathcal{D}_2$
Task m, $\mathcal{D}_m$
Environment
Unseen Task
Unseen Task
Unseen Task
Model-Driven
Prior Extraction via Learning
Data-Driven
Machine Learning Model (Implicit System Dynamics)
Model Conversion
SDE Representation (Explicit System Dynamics)
Kalman Filtering based Technique
Efficient Modeling: Data-Driven
Online Inference/Decision: Model-Driven

Fig. 1. The principle of hybrid model and data driven based DIIKF approach.

To train a ML system or prediction model, we first collect the training samples. For the BPT problem, the dataset can be collected online [15]. Then, we choose an appropriate learning model, and train the model to obtain an implicit representation of system dynamics. With the implicit representation of system dynamics available, we further convert the implicit representation into an explicit representation, i.e., a SDE that describes system evolution, so as to use the Kalman filter. Now, the first important issue is what ML model should we choose.

In practice, the system dynamics is almost always characterized via a system of SDEs, which can be written as

$$d\boldsymbol{x}(t) = \boldsymbol{f}(\boldsymbol{x})dt + \boldsymbol{\sigma}(\boldsymbol{x})d\xi(t), \tag{4}$$

where $\boldsymbol{x}(t)$ and $\xi(t)$ represent the system state and Brownian motion, respectively. In (4), $\boldsymbol{f}(\boldsymbol{x})$ and $\xi(t)$ (modulated by the volatility term $\boldsymbol{\sigma}(\boldsymbol{x})$) characterize the deterministic part and stochastic part of system evolution, respectively. It seems that we can obtain the system dynamics via estimating $\boldsymbol{f}(\boldsymbol{x})$ and $\boldsymbol{\sigma}(\boldsymbol{x})$. However, it is a difficult task, because the data samples for $\boldsymbol{f}(\boldsymbol{x}, t)$ and $\boldsymbol{\sigma}(\boldsymbol{x})$ are unavailable. To tackle this issue, we need the following theorem [20]. Note that a brief introduction of GP is provided in Appendix A.

Theorem 1. *There exist a Borel measurable function $\boldsymbol{F}$ and a basic GP (e.g., a GP with the SE kernel k_{SE}) such that the probability measure of the GP (with transform $\boldsymbol{F}(\cdot)$ as input) coincides with the probability measure of the SDE in* (4).

Theorem 1 shows that a "simple" ML model that incorporates the neural network and basic GP can completely characterize the probabilistic properties of dynamical system (4).

Remark 3.1 Theorem 1, in fact, constitutes the theoretical foundation of the DIIKF approach. First, as a bridge, it links two different branches of system modeling methodologies. In particular, it provides an efficient ML model/structure for the dynamical system, which avoids heuristic (and even) random selection of the prediction model. Moreover, it guarantees that the ML model does not cause any systematic error.

B. Data-Driven Implicit System Dynamics Learning

Based on Theorem 1, the ML model consists of two parts, i.e., a nonlinear transform and a basic GP. As an example, the GP with the SE kernel k_{SE} is chosen as the basic GP. As for the nonlinear transform, it is represented by a deep neural network with parameters (i.e., weights and biases) collected into Θ_{T}. The basic GP is denoted by $k_{\mathrm{SE}}(\cdot,\cdot|\Theta_{\mathrm{B}})$, where Θ_{B} collects the parameters of the GP kernel. Let $\boldsymbol{\Theta}=\{\Theta_{\mathrm{T}},\Theta_{\mathrm{B}}\}$. Now, the key problem is to learn the implicit representation of system dynamics from a dataset $\mathfrak{D}$ by optimizing $\boldsymbol{\Theta}$.

In contrast to most ML applications, the dataset $\mathfrak{D}$ contains m sub-datasets $\boldsymbol{S}_1,\cdots,\boldsymbol{S}_m$ of m tasks, i.e., $\mathfrak{D}=\{\boldsymbol{S}_1,\cdots,\boldsymbol{S}_m\}$. For example, each $\boldsymbol{S}_i$ corresponds to the beam trajectory of a vehicle (e.g., the i-th vehicle), and $\boldsymbol{S}_i$ takes the form

$$\boldsymbol{S}_i=\{(t_1,\mathbf{a}_1),(t_2,\mathbf{a}_2),\cdots,(t_{n_i},\mathbf{a}_{n_i})\}, \tag{5}$$

where t_i is the sampling time and $\mathbf{a}_i$ is the beam direction. For the uniform linear array, $\mathbf{a}_i$ is simplified as $\mathbf{a}_i=\psi_{t_i}$. We highlight that since the learning of implicit system dynamics belongs to multi-task learning category, it is problematic to extend the single-task learning methods directly to the multi-task case, because the data distributions (in essence, the system parameters) of different sub-datasets are different.

To tackle this issue, we reinterpret the generation of $\mathcal{D}$ from the perspective of stochastic process and hierarchical Bayes model, in view the fact that these different sub-datasets share many features, e.g., a similar trend. Specifically, $\{\boldsymbol{S}_i\}$ can be regarded as different realizations of a stochastic process, and the stochastic process is characterized by a set of parameters, which describe the similarities of different realizations.

1) Reinterpretation via Stochastic Process: Mathematically, each $\boldsymbol{S}_i$ can be regarded as the discrete sampling version of a continuous-time beam direction $f_i(t)$, while $f_i(t)$ is thought of as sampling from the underlying physical system.

2) Hierarchical Bayes Modeling: Instead of fixed values, a prior distribution, denoted by $\mathcal{P}(\boldsymbol{\Theta})$, is imposed on $\boldsymbol{\Theta}$. For simplicity, $\boldsymbol{\Theta}$ is assumed to be distributed as $\mathcal{N}(\mathbf{0},\sigma^2_{\boldsymbol{\Theta}}\mathbf{I})$, i.e., $\mathcal{P}=\mathcal{N}(\mathbf{0},\sigma^2_{\boldsymbol{\Theta}}\mathbf{I})$. Given $\mathfrak{D}$, it is sufficient to update the prior $\mathcal{P}(\boldsymbol{\Theta})$ into the posterior $\mathcal{Q}(\boldsymbol{\Theta})$, so as to match $\mathcal{D}$.

Note that the update of $\boldsymbol{\Theta}$ requires an optimization criterion. A natural criterion is that the obtained posterior can maximize the generalization performance, or equivalently, minimize the transfer-error. Based on this optimization criterion, we derive an effective posterior in the following theorem [20].

Theorem 2. *With the aim of minimizing the generalization or transfer error, an efficient posterior $\mathcal{Q}^\star(\boldsymbol{\Theta})$ is given by*

$$\mathcal{Q}^\star(\boldsymbol{\Theta}) \\ =\frac{\mathcal{P}(\boldsymbol{\Theta})\exp\left(\left(1+\sum_{i=1}^m n_i^{-1}\right)^{-1}\sum_{i=1}^m\frac{1}{n_i}Z(\boldsymbol{S}_i,\boldsymbol{\Theta})\right)}{\mathbb{E}_{\boldsymbol{\Theta}\sim\mathcal{P}}\left[\exp\left(\left(1+\sum_{i=1}^m n_i^{-1}\right)^{-1}\sum_{i=1}^m\frac{1}{n_i}Z(\boldsymbol{S}_i,\boldsymbol{\Theta})\right)\right]},$$

where $Z(\boldsymbol{S}_i,\boldsymbol{\Theta})$ denotes the marginal log-likelihood for $\boldsymbol{S}_i$ and $\boldsymbol{\Theta}$. In particular, for GP, $Z(\boldsymbol{S}_i,\boldsymbol{\Theta})$ is calculated as

$$Z(\boldsymbol{S}_i,\boldsymbol{\Theta})=-\frac{1}{2}\mathbf{y}_i^{\mathrm{T}}(\mathbf{C}_i+\sigma_i^2\mathbf{I})^{-1}\mathbf{y}_i- \\ \frac{1}{2}\log\det(\mathbf{C}_i+\sigma_i^2\mathbf{I})-\frac{n_i}{2}\log 2\pi. \tag{6}$$

See Appendix A for the definitions of matrix $\mathbf{C}_i$ and vector $\mathbf{y}_i$.

To facilitate the use of Kalman filter and reduce complexity, a point estimate of $\boldsymbol{\Theta}$ is preferable. The optimal point estimate is the MAP (maximum a posterior) estimate, i.e.,

$$\boldsymbol{\Theta}^\star=\arg\max_{\boldsymbol{\Theta}}\mathcal{Q}^\star(\boldsymbol{\Theta}). \tag{7}$$

Equivalently, we can also obtain the optimal estimate $\boldsymbol{\Theta}^\star$ more simply by maximizing the numerator of $\mathcal{Q}^\star(\Theta)$, i.e.,

$$\boldsymbol{\Theta}^\star=\arg\max_{\boldsymbol{\Theta}}\ln\mathcal{P}(\boldsymbol{\Theta})+\left(1+\sum_{i=1}^m\frac{1}{n_i}\right)^{-1}\sum_{i=1}^m\frac{Z(\boldsymbol{S}_i,\boldsymbol{\Theta})}{n_i}.$$

Once $\boldsymbol{\Theta}^\star$ is obtained, the implicit system dynamics has been determined, based on which Bayesian inference can be made when faced with new tasks. But, the inference is based on GP regression, which leads to large computational complexity.

C. SDE Representation and Efficient Inference

To enable efficient inference, explicit system dynamics, i.e., the SDE describing system state evolution, is required. For simplicity, we consider the one-dimensional case and denote the transform of the neural network by $z=F(t)$. We utilize the spectral factorization method [21] to convert the basic GP model obtained from $\boldsymbol{\Theta}^\star$ into an SDE representation.

1) Step 1: By computing the Fourier transform of the basic kernel (e.g., k_{SE}), we obtain the power spectral density $S(\omega)$.

2) Step 2: $S(\omega)$ can be written (or approximated via Taylor series expansion) as a rational function taking the form

$$S(\omega)=\frac{1}{p_n(\omega^2)}, \tag{8}$$

where $p_n(\cdot)$ represents a polynomial of nth order.

3) Step 3: A stable rational transfer function $H(i\omega)$ can be found, which takes the form

$$H(i\omega)=\frac{\sigma_0}{(i\omega)^n+a_{n-1}(i\omega)^{n-1}+\cdots+a_1(i\omega)+a_0}. \tag{9}$$

Then, $S(\omega)$ can be written as $S(\omega)=\sigma_0^2H(i\omega)H(-i\omega)$. [1]

[1] The procedure to find the transfer function is called spectral factorization, which consists of two steps. First, the roots (always appearing in pairs) of the denominator are computed. Then, the denominator polynomial of $H(i\omega)$ can be constructed from the positive-imaginary-part roots only.

With the transfer function $H(j\omega)$ available, we can prove that the desired SDE can be written as

$$\frac{d^n u(z)}{dz^n} + \cdots + a_1 \frac{du(z)}{dz} + a_0 u(z) = \sigma_0 \frac{d\xi(z)}{dz}.$$

Let $\boldsymbol{x}(z) = (u(z), du(z)/dz, \cdots, d^{n-1}u(z)/dz^{n-1})^{\mathrm{T}}$ and $\mathbf{v} = (0, \cdots, 0, \sigma_0)^{\mathrm{T}}$. The SDE can be rewritten as

$$d\boldsymbol{x}(z) = \mathbf{C}\boldsymbol{x}(z)dz + \mathbf{v}d\xi(z), \tag{10}$$

where matrix $\mathbf{C}$ is given by

$$\mathbf{C} = \begin{pmatrix} 0 & 1 & & \\ & \ddots & \ddots & \\ & & 0 & 1 \\ -a_0 & -a_1 & \cdots & -a_{n-1} \end{pmatrix}.$$

Let $\mathbf{b} = [1, 0, \cdots, 0]^{\mathrm{T}}$ whose dimension matches with $\boldsymbol{x}(t)$. Then, the measurement equation can be written as

$$b(t) = \mathbf{b}^{\mathrm{T}}\boldsymbol{x}(z(t)) + \eta(t),$$

where $\eta(t)$ denotes the measurement noise.

In practice, only discrete-time observations can be obtained. Let t_k and e_k with $e_k \sim \mathcal{N}(0, \sigma_{\mathrm{m}}^2)$ $(k = 1, 2, \cdots)$ denote the sampling time and observation noise, respectively. Under the assumption that the system parameters keep constant within each time-slot, we can obtain the discrete-time model:

$$\boldsymbol{x}(z_{k+1}) = (\mathbf{I} + \mathbf{C}\Delta z_k)\boldsymbol{x}(z_k) + \mathbf{q}_k \tag{11}$$

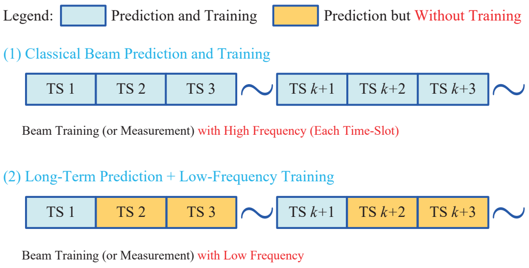
$$b(t_k) = \mathbf{b}^{\mathrm{T}}\boldsymbol{x}(z_k) + e_k, \tag{12}$$

where $\mathbf{q}_k$ stemming from the approximation is assumed to be Gaussian. Under the assumption that $\Delta z_k = F(t_{k+1}) - F(t_k)\,(\forall\, k)$ is small, the classical discrete-time KF algorithm can be invoked, with an appropriate initial condition $\boldsymbol{x}(z_0) \sim \mathcal{N}(\boldsymbol{x}_0, \mathbf{P}_0)$ [20]. For clarity, the complete procedure of the proposed DIIKF approach is summarized in Algorithm 1.

```
Algorithm 1: Data-Induced Intelligent Kalman Filtering
1: input: training datasets $\mathfrak{D} = \{\boldsymbol{S}_1, \boldsymbol{S}_2, \cdots, \boldsymbol{S}_m\}$
2: initialize the neural network $F$ with $\Theta_{\mathrm{T}}$ and GP with $\Theta_{\mathrm{k}}$
3: repeat
       (a) sample a small batch of sub-datasets from $\mathfrak{D}$
       (b) compute training loss with the sampled data
       (c) update $\{\Theta_{\mathrm{T}}, \Theta_{\mathrm{k}}\}$ the gradient descent method
   until convergence condition is met $\Longrightarrow \boldsymbol{\Theta}^\star$
4: convert GP representation into SDE representation
5: generate: nonlinear transform $F$ and state-space equations
6: predict online and recursively: (1) $t_k$ is fed into $F$;
       (2) predict beam direction via KL with $F(t_k)$ as input
```

IV. Low-Frequency Beam Training via Long-Term Prediction

In the classical BPT scheme, beam measurement (i.e., local beam sweeping) is executed in each time-slot, as shown in Fig. 2-(1). The high frequency of beam measurement degrades system performance of interest (e.g., effective achievable rate). In view that the SDE characterizing the variation tendency of the underlying physical system is available, we next propose to predict its long-term variation behavior. As shown in Fig. 2-(2), low frequency of beam measurement is involved in the new BPT scheme, which further improves the performance.



Legend:
Prediction and Training
Prediction but Without Training
(1) Classical Beam Prediction and Training
TS 1
TS 2
TS 3
TS k+1
TS k+2
TS k+3
Beam Training (or Measurement) with High Frequency (Each Time-Slot)
(2) Long-Term Prediction + Low-Frequency Training
TS 1
TS 2
TS 3
TS k+1
TS k+2
TS k+3
Beam Training (or Measurement) with Low Frequency

Fig. 2. The comparison of the classical beam prediction (and training) scheme and the novel long-term beam prediction (and training) scheme.

Note that because $F(\cdot)$ is almost always a nonlinear transform and Δt may span across multiple time-slots in the long-term BPT scheme. As a result, the discretization in (11) and (12) could be problematic in this case. To tackle this issue, we propose an efficient discretization method below.

Lemma 1 ([20]). *The SDE in* (10) *is in distribution equivalent to the following discrete-time dynamical system* [2]

$$\boldsymbol{x}(z_{k+1}) = \mathbf{T}_k\boldsymbol{x}(z_k) + \mathbf{q}_k, \qquad \mathbf{q}_k \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Pi}_k), \tag{13}$$

where $\mathbf{T}_k$ *and* $\boldsymbol{\Pi}_k$ *are respectively given by*

$$\mathbf{T}_k = \exp(\mathbf{C}(z_{k+1} - z_k))$$

$$\boldsymbol{\Pi}_k = \int_{z_k}^{z_{k+1}} \exp(\mathbf{C}(z_{k+1} - s))\mathbf{v}\mathbf{v}^{\mathrm{T}}\exp(\mathbf{C}(z_{k+1} - s))^{\mathrm{T}}ds.$$

With Lemma 1 available, we can characterize the predictive distribution of beam direction $b(t)$. Without loss of generality, we focus on an arbitrary but fixed time t_k and predict beam $b(t)$ for $t > t_k$. The first-order and second-order statistics of the predictive distribution are presented in Theorem 3 [20].

Theorem 3. *The predictive distribution of* $b(t)$ *is given by*

$$b(t) \sim \mathcal{N}(m(t), a(t)), \quad (t > t_k), \tag{14}$$

where $m(t) = \mathbf{b}^{\mathrm{T}}\mathbf{T}(F(t_k), F(t))\boldsymbol{x}(F(t_k))$ *and* $a(t) = \sigma_m^2 + \mathbf{b}^{\mathrm{T}}\boldsymbol{\Pi}(F(t_k), F(t))\mathbf{b}$, *with* $\mathbf{T}(z_k, z)$ *and* $\boldsymbol{\Pi}(z_k, z)$ *given by*

$$\mathbf{T}(z_k, z) = \exp(\mathbf{C}(z - z_k))$$

$$\boldsymbol{\Pi}(z_k, z) = \int_{z_k}^{z} \exp(\mathbf{C}(z - s))\mathbf{v}\mathbf{v}^{\mathrm{T}}\exp(\mathbf{C}(z - s))^{\mathrm{T}}ds.$$

The variance $a(t)$ is a monotone increasing function, which implies that the predicted beam direction becomes more inaccurate as t increases. If $a(t)$ is relatively large (e.g., it is greater than a predefined threshold value), we shall redetermine the optimal beam. For example, we can find out the optimal beam via the hierarchical search [2] or sweep the beam interval $(m(t) - c\sqrt{a(t)}, m(t) + c\sqrt{a(t)})$, where $c > 0$ is a constant.

[2] The equivalence means that the probability distributions of the continuous-time SDE and the discrete-time system coincide at the sampling points $\{t_k\}$.

V. Numerical Results

In this section, simulation results are provided to demonstrate the performance of the proposed algorithms. Without loss of generality, the uniform linear array is considered. Two cases of antenna array ($N = 64$ and $N = 128$) are chosen to evaluate different algorithms. The size of codebook $\mathcal{C}$ satisfies $M = N$ (with accuracy $1/N$). For all experiments, the channel model in (1) includes one LOS path and three NLOS paths. The angles of the NLOS paths are distributed uniformly in $[0, 2\pi)$. The average power ratio of the LOS path gain and each NLOS path gain is set to 10dB. A uniform distribution with value domain $[a, b]$ is denoted by $\mathrm{U}(a, b)$. The simulation setting and key system parameters are described below.

Two road conditions are chosen in this paper to evaluate different algorithms, where one is a straight road (Road 1) and the other one is a cycloid-like road (Road 2). The length of each time-slot (i.e., transmission time interval) is set to 20 milliseconds. The width of Road 1 is 20m, while the width of Road 2 is 15m. It is allowed to change lane in Road 1, with probability 1/10. The initial velocity (i.e., the velocity when the vehicle enters the coverage area) in Road 1 is distributed uniformly as $\mathrm{U}(54, 108)$ (km/h), while the initial velocity in Road 2 is distributed as $\mathrm{U}(54, 80)$ (km/h). With probability 0.1, the driver speeds up (or slows down) the velocity to the maximal (or minimal) speed in each road.

We compare our methods to the classical hierarchical search (HS) based method [2] and the state-of-the-art ML algorithms (i.e., the stochastic bandit learning (SBL) based algorithm [14] and direct upper confidence bound (DUCB) based algorithm). The oracle aided algorithm is served as a benchmark, which can always find out the optimal beam with training overhead zero. For convenience, the BPT algorithm incorporating only implicit system dynamics learning along with GP inference, BPT algorithm with DIIKF only (i.e., Algorithm 1) and BPT solution incorporating both DIIKF and long-term prediction are named as ISDL+GPI, DIIKF and DIIKF+LTP, respectively. The average effective achievable rate (AEAR) and probability of successful alignment (PSA) are chosen as the performance metrics to evaluate different BPT algorithms.

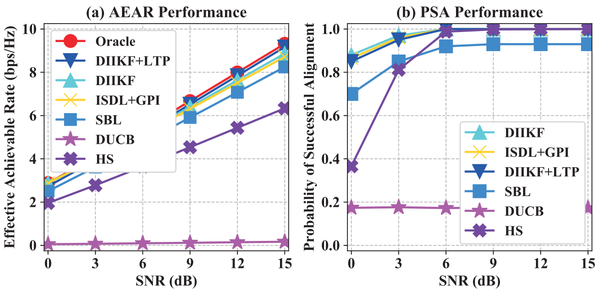


Fig. 3. The AEAR and PSA performance of different algorithms - $N = 64$.

Fig. 3 demonstrates the AEAR and PSA performance. It is observed that the algorithms proposed in this paper outperform the other BPT algorithms. In particular, DIIKF+LTP can even approach the oracle-aided BPT solution. The reason for this is that the underlying system dynamics is efficiently learned, which can effectively guide the long-term prediction. As a result, the overhead of beam training can be greatly reduced, and meanwhile the optimal beams can still be correctly tracked. It is also observed that DUCB achieves the worst performance in terms of both AEAR and PSA. The reason for this is that it is mainly applicable for the slow-varying environment.

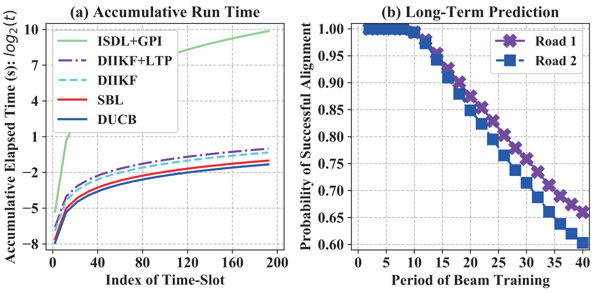


Fig. 4. The run-time and long-time prediction performance - $N = 128$.

An appealing advantage of our approach is that it inherits the property of low computational complexity from the classical KF technique. Thanks to the linear complexity order, it is not surprising that the curves (in terms of accumulative run-time) of DIIKF and DIIKF+LTP increase very slow, as shown in Fig. 4-(a). In contrast, due to the cubic complexity order, the time-resource required by ISDL+GPI increases remarkably. Still thanks to the KF characteristics, DIIKF can well adapt to the non-stationary environments, which accounts for a little bit better performance than ISDL+GPI. Fig. 4-(b) intuitively demonstrates the long-term performance of DIIKF+LTP. Since a prediction toward longer future introduces larger uncertainty, the PSA performance accordingly decreases.

VI. Conclusion

To exploit the advantages of both KF and ML, we proposed a novel hybrid model and data driven approach, referred to as DIIKF. First, we theoretically derived an efficient learning or model structure, which lays the foundation for DIIKF. Then, we proposed an effective method to learn the implicit system dynamics. To avoid frequent beam training, we also proposed an efficient long-term prediction based BPT algorithm.

Appendix A
Gaussian Process Regression

A stochastic process $f(\mathbf{x})$ is referred to as a GP if and only if for any finite number of points $\mathbf{x}_1, \cdots, \mathbf{x}_n$ ($\forall\, i, \mathbf{x}_i \in \mathbb{R}^d$), the joint probability density function $p(f(\mathbf{x}_1), \cdots, f(\mathbf{x}_n))$ is Gaussian [22]. A GP is completely characterized by the mean function $m(\mathbf{x})$ and covariance function $k(\mathbf{x}, \mathbf{x}')$, which are similar to the mean and covariance for a Gaussian vector. The mean and covariance functions are respectively defined as

$$\begin{aligned} m(\mathbf{x}) =& \mathbb{E}[f(\mathbf{x})], \\ k(\mathbf{x}, \mathbf{x}') =& \mathbb{E}\big[\big(f(\mathbf{x}) - m(\mathbf{x})\big)\big(f(\mathbf{x}') - m(\mathbf{x}')\big)\big]. \end{aligned} \tag{15}$$

The mean function is assumed to be zero next, i.e., $m(\mathbf{x}) = 0$.

GP regression is to predict or infer $f(\mathbf{x}_\star)$ for an unseen $\mathbf{x}_*$ based on a set of observations $\boldsymbol{S} = \{(\mathbf{x}_i, y_i) \,|\, y_i = f(\mathbf{x}_i) + w_i, w_i \sim \mathcal{N}(0, \sigma^2), i = 1, \cdots, n\}$, where $\mathbf{x}_i \in \mathcal{X} \subset \mathbb{R}^d$ and $y \in \mathbb{R}$ denote input vector and output label, respectively. In contrast to many parametric regression methods, GP regression is based on Bayesian inference, which generates a probability distribution, rather than only a point estimate for the quantity of interest. Given $\boldsymbol{S}$ above, we can next derive the conditional or predictive distribution for $f_* = f(\mathbf{x}_*)$ at $\mathbf{x}_*$ [22].

The observed points are stacked into $\mathbf{y}_{\mathrm{o}} = (y_1, \cdots, y_n)^{\mathrm{T}}$. Based on the Gaussian assumption, the joint probability distribution between $\mathbf{y}_{\mathrm{o}}$ and $y_* = f(\mathbf{x}_*) + w_*$ is given by

$$\begin{bmatrix} \mathbf{y}_{\mathrm{o}} \\ y_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{pmatrix} \mathbf{C}_{\mathrm{oo}} + \sigma^2 \mathbf{I} & \mathbf{c}_{*\mathrm{o}} \\ \mathbf{c}_{*\mathrm{o}}^{\mathrm{T}} & c_{**} \end{pmatrix} \right), \tag{16}$$

where the matrices/vectors are formed as $[\mathbf{C}_{\mathrm{oo}}]_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$, $[\mathbf{c}_{*\mathrm{o}}]_j = k(\mathbf{x}_*, \mathbf{x}_j)$ and $c_{**} = k(\mathbf{x}_*, \mathbf{x}_*)$. Then, the conditional distribution of $f_* = f(\mathbf{x}_*)$ at $\mathbf{x}_*$ is given by

$$p(f_* | \boldsymbol{S}, \mathbf{x}_*) \sim \mathcal{N}\big(\mu(\mathbf{x}_*), c(\mathbf{x}_*)\big) \tag{17}$$

$$\mu(\mathbf{x}_*) = \mathbf{c}_{*\mathrm{o}}^{\mathrm{T}} (\mathbf{C}_{\mathrm{oo}} + \sigma^2 \mathbf{I})^{-1} \mathbf{y}_{\mathrm{o}} \tag{18}$$

$$c(\mathbf{x}_*) = c_{**} - \mathbf{c}_{*\mathrm{o}}^{\mathrm{T}} (\mathbf{C}_{\mathrm{oo}} + \sigma^2 \mathbf{I})^{-1} \mathbf{c}_{*\mathrm{o}}. \tag{19}$$

The GP kernel is the crucial ingredient for a GP predictor, because it encodes the prior knowledge about the function to be learned. The classical squared exponential (SE) kernel is chosen in this paper, which takes the form

$$k_{\mathrm{SE}}(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left(-\frac{1}{2l^2} \|\mathbf{x} - \mathbf{x}'\|^2 \right), \tag{20}$$

where σ_f^2 and l represent the signal variance and length-scale, respectively. The physical meaning of parameter l is that if the GP varies rapidly, the length-scale l should be shorter [22]. Hence, the degree of GP variation can be achieved by adjusting the parameters. It is referred to [22] for more details.

Appendix B
Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grants 62301249, 61720106003 and 62225107, the Foundation Research Project of Jiangsu Province under Grant BK20230878, the Major Key Project of PCL (PCL2021A01-2), and the Engineering and Physical Sciences Research Council, UK under project EP/S028455/1.

References

[1] M. Xiao, S. Mumtaz, Y. Huang, L. Dai, Y. Li, M. Matthaiou, G. K. Karagiannidis, E. Björnson, K. Yang, C. L. I, and A. Ghosh, "Millimeter wave communications for future mobile networks," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 9, pp. 1909–1935, Sept 2017.

[2] S. Hur, T. Kim, D. Love, J. Krogmeier, T. Thomas, and A. Ghosh, "Millimeter wave beamforming for wireless backhaul and access in small cell networks," *IEEE Trans. Commun.*, vol. 61, no. 10, pp. 4391–4403, October 2013.

[3] J. Singh and S. Ramakrishna, "On the feasibility of codebook-based beamforming in millimeter wave systems with multiple antenna arrays," *IEEE Trans. Wireless Commun.*, vol. 14, no. 5, pp. 2670–2683, May 2015.

[4] Z. Xiao, T. He, P. Xia, and X. G. Xia, "Hierarchical codebook design for beamforming training in millimeter-wave communication," *IEEE Trans. Wireless Commun.*, vol. 15, no. 5, pp. 3380–3392, May 2016.

[5] F. Liu, W. Yuan, C. Masouros, and J. Yuan, "Radar-assisted predictive beamforming for vehicular links: Communication served by sensing," *IEEE Trans. Wireless Commun.*, vol. 19, no. 11, pp. 7704–7719, 2020.

[6] S. G. Larew and D. J. Love, "Adaptive beam tracking with the unscented kalman filter for millimeter wave communication," *IEEE Signal Process. Lett.*, vol. 26, no. 11, pp. 1658–1662, 2019.

[7] W. Yuan, F. Liu, C. Masouros, J. Yuan, D. W. K. Ng, and N. Gonzlez-Prelcic, "Bayesian predictive beamforming for vehicular networks: A low-overhead joint radar-communication approach," *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp. 1442–1456, 2021.

[8] V. Va, J. Choi, T. Shimizu, G. Bansal, and R. W. Heath, "Inverse multipath fingerprinting for millimeter wave V2I beam alignment," *IEEE Trans. Veh. Technol.*, vol. 67, no. 5, pp. 4042–4058, May 2018.

[9] K. Satyanarayana, M. El-Hajjar, A. A. M. Mourad, and L. Hanzo, "Deep learning aided fingerprint-based beam alignment for mmwave vehicular communication," *IEEE Trans. Veh. Technol.*, vol. 68, no. 11, pp. 10 858–10 871, Nov 2019.

[10] V. Va, T. Shimizu, G. Bansal, and R. W. Heath, "Online learning for position-aided millimeter wave beam training," *IEEE Access*, vol. 7, pp. 30 507–30 526, 2019.

[11] M. Cheng, J. Wang, J. Wang, M. Lin, Y. Wu, and H. Zhu, "A fast beam searching scheme in mmwave communications for high-speed trains," in *2019 IEEE ICC*, May 2019, pp. 1–6.

[12] W. Wu, N. Cheng, N. Zhang, P. Yang, W. Zhuang, and X. Shen, "Fast mmwave beam alignment via correlated bandit learning," *IEEE Trans. Wireless Commun.*, vol. 18, no. 12, pp. 5894–5908, 2019.

[13] M. B. Booth, V. Suresh, N. Michelusi, and D. J. Love, "Multi-armed bandit beam alignment and tracking for mobile millimeter wave communications," *IEEE Commun. Lett.*, vol. 23, no. 7, pp. 1244–1248, 2019.

[14] J. Zhang, Y. Huang, Y. Zhou, and X. You, "Beam alignment and tracking for millimeter wave communications via bandit learning," *IEEE Trans. Commun.*, vol. 68, no. 9, pp. 5519–5533, 2020.

[15] J. Zhang, Y. Huang, J. Wang, X. You, and C. Masouros, "Intelligent interactive beam training for millimeter wave communications," *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp. 2034–2048, 2021.

[16] J. Zhang and C. Masouros, "Learning-based predictive transmitter-receiver beam alignment in millimeter wave fixed wireless access links," *IEEE Trans. Signal Process.*, vol. 69, pp. 3268–3282, 2021.

[17] J. Zhang, W. Xu, H. Gao, M. Pan, Z. Feng, and Z. Han, "Position-attitude prediction based beam tracking for UAV mmwave communications," in *2019 IEEE ICC*, 2019, pp. 1–7.

[18] H.-L. Song and Y.-C. Ko, "Beam alignment for high-speed UAV via angle prediction and adaptive beam coverage," *IEEE Trans. Veh. Technol.*, vol. 70, no. 10, pp. 10 185–10 192, 2021.

[19] A. Alkhateeb, S. Alex, P. Varkey, Y. Li, Q. Qu, and D. Tujkovic, "Deep learning coordinated beamforming for highly-mobile millimeter wave systems," *IEEE Access*, vol. 6, pp. 37 328–37 348, 2018.

[20] J. Zhang, H. Yongming, C. Masouros, and X. You, "Data-induced intelligent Kalman filtering approach and its application to wireless communications," *Submitted to IEEE Trans. Signal Process.*, 2023.

[21] S. Sarkka, A. Solin, and J. Hartikainen, "Spatiotemporal learning via infinite-dimensional bayesian filtering and smoothing: A look at gaussian process regression through kalman filtering," *IEEE Signal Process. Mag.*, vol. 30, no. 4, pp. 51–61, 2013.

[22] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2006.