# Machine Learning With Gaussian Process Regression For Time-Varying Channel Estimation

Richard Simeon, Taejoon Kim, and Erik Perrins
Department of Electrical Engineering & Computer Science
University of Kansas; Lawrence, KS 66045
E-mails: {rsimeon, taejoonkim, esp}@ku.edu

***Abstract*—The minimum mean-squared error (MMSE) estimator is recognized as the best estimator for measuring transmission channel distortion in orthogonal frequency division multiplexing (OFDM) using pilot-symbol assisted modulation (PSAM) in the presence of noise. In practice, however, the estimator suffers from high complexity and relies on the estimation of second-order statistics which may change rapidly within small-scale fading environments in a high-mobility wireless transmission system. We propose using machine learning (ML) with Gaussian Process Regression (GPR) to adaptively learn the hyperparameters of a channel model, which then can be used to calculate the MMSE estimates. Moreover, GPR can be used to more accurately interpolate the channel estimates in between pilot symbols compared to linear interpolation techniques. After describing the learning process and its equivalency to MMSE, we derive the BER for a receiver using GPR for time-domain interpolation, then use BER to find a practical bound on the number of training points needed to achieve best performance. We show that the performance of GPR-based ML is comparable to that of more complex neural network-based ML.**

***Index Terms*—Machine learning, channel estimation, Gaussian process, OFDM**

## I. Introduction

The next-generation high-mobility communications systems beyond-5G (B5G) are targeting high-speed railway (HSR) use cases requiring reliable data rates of at least 100 Mbps at peak speeds of 500 kph [1] with low power consumption. Comparatively, earlier 4G LTE (Long-Term Evolution) systems guaranteed only functional (low bit rate) services above 120 kph [2]. To achieve B5G goals of high performance, high-mobility, and low power, accurate channel estimation is needed for mobile terminals requiring minimum signaling overhead, adaptation to fast-changing channel statistics due to terminal mobility, and computational compactness for low power consumption.

Channel state information (CSI) is needed to undo the effects of channel distortion and decode error-free received symbols; this can be achieved via pilot-symbol assisted modulation (PSAM), where measured observations of known pilot (training) signals periodically-transmitted in between information symbols of synchronized transmission systems are used to estimate the CSI [3]. New channel estimation algorithms are needed to be able to adapt quickly when channel statistics change due to mobile velocity changes in multi-path environments, and must be robust enough to learn the channel accurately when pilots are spaced far apart relative to the channel coherence time.

Deep learning for wireless communications systems has recently received plenty of research interest due to the generalized ability to learn unknown or complex channel models and nonlinearities, and the realizations made possible by modern computing architectures [4]. However, limited research has been done to date on the effects of mobility on learned systems.

### A. Prior Work and Motivation

Deep learning for digital receivers has been investigated in [5] using deep neural networks (DNNs) to combine channel estimation with symbol detection and achieve near-minimum mean-squared error (MMSE) performance, but did not consider time variation in the channel. In [6] a signal preamble is used for supervised learning of a slow time-varying channel, with channel estimates between pilots linearly interpolated. In [7] the piecewise linear properties of fully-connected ReLU (rectified linear unit) DNNs are used to show the ability to approximate MMSE estimators.

DNNs are trained offline using large amounts of labeled channel measurement data to ensure adequate convergence. In many practical application scenarios, however, we may not have or it is too expensive to obtain a large amount of reliably labeled data. For channel estimation, this typically constrains the DNN solution to static channels, with limited support for mobility. DNNs also lack interpretability of the trained model; after training, the weights of the neurons cannot be readily analyzed to interpret the characteristics of the channel. This contributes to an open problem of being unable to analytically characterize performance to bound the number of layers and neurons needed to accurately estimate CSI. Generally it cannot be explained why performance is better or worse due to the inability to interpret how the models compensate for distortions that traditional models cannot correct [8].

Rather than burden the DNN with having to learn the channel with no prior statistical knowledge, model-based machine learning takes a parameterized approach, distilling the amount of learning to a handful of hyperparameters. This

T. Kim was supported in part by the National Science Foundation (NSF) under Grants CNS1955561 and the Office of Naval Research (ONR) under Grant N00014-21-1-2472.

allows for faster training, which lends itself to fast time-varying channels where channel states change quickly. It also allows for interpretability, since observation of the model hyperparameters can give insight to the underlying channel characteristics.

### *B. Overview of Methodology and Contributions*

We propose a model-based ML approach to channel estimation in order to quickly train to fast-changing channel conditions more efficiently than DNN-based machine learning approaches. Gaussian Process Regression (GPR) is a non-parametric Bayesian approach towards regression problems that uses a kernel to characterize the proposed model. It is especially useful when the phenomena to be estimated can be closely characterized by a Gaussian process [9]. Jakes [10] characterized small-scale channel fading in terms of a Gaussian process; as such, we use this as motivation to use GPR as an ideal technique for supervised model-based machine learning of channel estimates using PSAM pilots.

In addition to being interpretable due to the model-based approach, GPR is robust to low signaling overhead, which can happen if pilots are spaced far apart in time relative to the channel coherence time. The smaller number of hyperparameters to train lends itself to lower computational complexity, as opposed to the large amounts of neurons to train per pilot symbol. Finally, as will be presented, GPR fully establishes a mean and variance for each channel estimate, which means performance is predictable; computational complexity can be further reduced to analytically to meet a defined performance criteria.

Our key contributions are summarized as follows:

- We introduce GPR and show that the optimal kernel for multipath fading to achieve near-MMSE performance on pilot channel estimates is based on the Bessel function. We also identify the ability to obtain channel estimates of information symbols using the trained GPR[1].
- We analyze bit error rate (BER) performance of a digital receiver using GPR to find a reasonable bound on the number of training points needed to achieve the asymptotic performance bound.
- We present simulation results comparing GPR to a trained deep neural network and show that performance of GPR is comparable to the more-complex DNN.

*Notation:* A bold lower case $\mathbf{a}$ is a column vector, $a$ is a scalar, a bold upper case $\mathbf{A}$ is a matrix, $\mathbf{a}[i]$ is the $i$th entry of $\mathbf{a}$, $\mathbf{A}(i)$ is the $i$th column, and $\mathbf{A}(i,j)$ or $\mathbf{A}_{ij}$ is the $i$th row and $j$th column entry of $\mathbf{A}$. $\mathbf{I}_M \in \mathbb{R}^{M\times M}$ is the identity matrix. tr($\mathbf{A}$), $\mathbf{A}^T$, and $\mathbf{A}^H$ are, respectively, the trace, transpose, and conjugate transpose of $\mathbf{A}$. The estimate of $a$ is $\hat{a}$, and the absolute value of $a$ is $|a|$. $\mathbb{E}[\cdot]$ denotes the expectation operator.

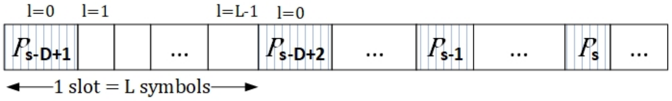

Fig. 1. PSAM slot structure with pilot and information symbols.

## II. System Model

### *A. Pilot Symbol Aided Modulation (PSAM)*

In the presence of system noise, channel distortion can be estimated with pilot symbols known *a priori*, and the effects of that distortion is undone in the receiver for optimal information symbol detection. Periodic pilot transmission allows for the receiver to adjust for time-variations in the channel.

We adopt the LTE convention of a "slot" and define data transmission as a series of slots of length $L$ symbols with symbol duration $T$, each slot containing a pilot symbol at slot position $l = 0$ and data symbols at slot positions $l = 1, \ldots L-1$ [11]. We show this in Fig. 1, where $p_s$ is the pilot at current slot $s$, and the last $D$ pilot symbols are shown.

The pilot symbols must replace data symbols, which consequently decreases the system data throughput by a factor of $L^{-1}$. It is advantageous to transmit pilots as infrequently as possible, whilst balancing the need for fresh channel estimates as the channel characteristics change. Channel estimates for the information symbols are obtained via interpolation between pilots. Simple interpolation techniques may not hold if the channel changes faster than the pilots are transmitted when mobile speeds increase. Accurate interpolation is therefore needed for optimal decoding, and we show later that GPR satisfies this need.

### *B. Signal Model*

Referring back to Fig. 1, let $z = sL + l$ be the $z$-th symbol index. Fig. 2 shows the baseband narrowband communications model for analysis, with $a_z \in \mathbb{C}$ being the $z$-th transmitted symbol sourced from either information symbols $d_z \in \mathbb{C}$ for $l = 1..L-1$ or pilot symbols $p_s \in \mathbb{C}$ for slot $s$ when $l = 0$ ($\mathbb{E}[a_z] = 0$), $h_z \in \mathbb{C}$ the $z$-th symbol's channel gain (distortion), $q_z \in \mathbb{C}$ the corresponding receiver input signal, and $n_z \sim \mathcal{CN}(0, \sigma_n^2)$ is the observation noise consisting of additive independent and identically distributed (i.i.d.) Gaussian noise with zero mean and variance $\sigma_n^2$. For ease of exposition, we assume the channel is narrowband and frequency-independent, in which this narrowband assumption can be easily extended to a subcarrier of orthogonal frequency division multiplexing (OFDM) used in 4G LTE systems and beyond. We also assume that frequency offset from Doppler is removed by the receiver's automatic frequency correction (AFC), such that only residual small-scale fading effects remain. For notational simplicity, we omit the symbol index $z$ unless needed for clarity.

In Fig. 2, the channel estimate $\hat{h} \in \mathbb{C}$ is applied to the received input signal (equalization) via zero-forcing detection to recover the information signal $a$. The complex-valued

[1] While this paper focuses on time-domain interpolation of channel estimates, extension of GPR with the Bessel kernel to doubly-selective channels (time- and frequency-domain selectivity) in OFDM is straightforward so long as the kernel design introduced later in this paper is selected judiciously to suit the frequency-domain correlation statistics.

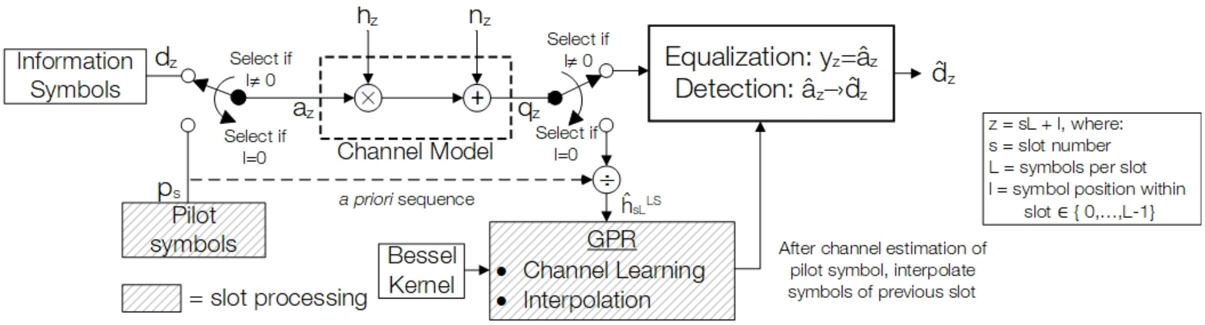

Fig. 2. Signal and System Model

received signal $y$ is then

$$y = \frac{ah+n}{\hat{h}} = \hat{a}, \tag{1}$$

where the channel estimate is decomposed into $\hat{h} = h + \delta h$, and $\delta h$ denotes the channel estimation (mismatch) error. The equalized signal can also be decomposed into the original signal $a$ and its distortion $\delta a$, so $y = (a + \delta a) + (n/\hat{h})$. Solving for $\delta a$,

$$\delta a = -a\delta h/(h + \delta h). \tag{2}$$

This results in an expression that links the channel estimation error with the received error.

### C. Mobile Channel Characteristics

A moving mobile terminal has a maximum Doppler frequency $f_m = vf_c/c$, where $f_c$ is the carrier frequency of the transmission, $v$ is the mobile velocity with respect to the transmitter, and $c$ is the speed of light ($c = 3 \times 10^8$ m/s). To incorporate symbol transmission rate into mobile BER analysis, we introduce the normalized Doppler spread $f_m T$. The consequence of a high Doppler frequency manifests itself in the rapidity of the channel fading. We can quantitatively measure this effect by calculating the time that the channel is relatively stationary, known as the coherence time ($T_C$). Using the definition of 50% coherence time [12], $T_C$ is given by $T_C = 0.75/(f_m\sqrt{\pi})$.

In the case of small-scale fading in a mobile multi-path environment, the channel gain $h$ is a random variable with statistics governed by the particular fading environment (traditionally represented by a Rayleigh or Ricean distribution). The Rayleigh fading model is shown to be a Gaussian process as described by Jakes [10]. Assuming wide-sense stationarity, the time autocorrelation of the channel gain $h$ with respect to the time lag $\tau$ is given as

$$r_{hh}(\tau) = \mathbb{E}[h_z h_{z+\tau}^*] = PJ_0(2\pi f_m \tau), \tag{3}$$

where $P$ is the large-scale path loss and $J_0(\cdot)$ is the zeroth order Bessel function of the first kind.

## III. LS and MMSE Estimation

Least-squares (LS) estimation attempts to minimize the cost function

$$\hat{h}^{LS} = \arg\min_{\hat{h}} |q - \hat{h}a|^2 = \frac{q}{a}. \tag{4}$$

LS is by far the least complex to implement, requiring only a simple division per channel estimate.

Unlike the LS estimate that merely zero-forces the observation to recover the channel, the linear MMSE (LMMSE) estimate exploits ensemble statistics existing in the observed samples. By collecting the samples in a block with length $D$, the LMMSE estimate can be given by [13]

$$\hat{h}^{LMMSE} = \mathbf{r}_{hh}^H(\mathbf{R}_{hh} + \sigma_n^2 \mathbf{I}_D)^{-1}\hat{\mathbf{h}}^{LS}, \tag{5}$$

where the vector $\hat{\mathbf{h}}^{LS} = [\hat{h}_{sL}^{LS}, \hat{h}_{(s-1)L}^{LS}, \ldots, \hat{h}_{(s-D+1)L}^{LS}]^T \in \mathbb{C}^{D\times 1}$ is obtained by collecting the $D$ LS channel estimates in (4), $\mathbf{R}_{hh} = \mathbb{E}[\mathbf{h}\mathbf{h}^H] \in \mathbb{C}^{D\times D}$ is the auto-correlation matrix of the channel coefficients $\mathbf{h} = [h_z, h_{z-L}, \ldots, h_{z-(D-1)L}]^T \in \mathbb{C}^{D\times 1}$, and $\mathbf{r}_{hh} \in \mathbb{C}^{D\times 1}$ is the first column of $\mathbf{R}_{hh}$. The LMMSE estimator in (5) can be viewed as applying a linear filtering matrix to the LS estimate. The dilemma here is that $\mathbf{R}_{hh}$ cannot be known with certainty without knowledge of $h$.

## IV. Using Machine Learning with Gaussian Process Regression

### A. Gaussian Process Regression

GPR is a Bayesian learning technique that estimates a latent regression function $g(\cdot)$ from a set of noisy training measurements

$$v_i = g(u_i) + n_i \tag{6}$$

to perform the prediction when new data (observations) are available [9]. For instance, the $v_i$ in (6) can be viewed as the $i$-th channel estimate (e.g., either (4) or (5)) given the pilot input (training data) and the additive observation noise $n_i$ following the model in Fig. 2. The $g(\cdot)$ in (6) denotes the GPR model following a joint Gaussian distribution (a Gaussian process), and produces estimates with a Gaussian-distributed mean and variance (uncertainty) for each estimate. A kernel function describes a "best guess" on the underlying model that characterizes the phenomena to be estimated. The kernel comprises hyperparameters that are trained with training points to allow for multiple degrees-of-freedom in the model for maximizing the probability that the model matches the latent function as best as possible. Once hyperparameters are trained, the kernel is then used in the GPR process to formulate an estimate at any test point.

We define a training data set of $D$ training samples to be $\mathcal{S} = \{(u_i, v_i), i = 1, ..., D\}$, $u_i \in \mathbb{R}$ and $v_i \in \mathbb{C}$. We define a GP kernel function, $k(\cdot)$, as a covariance $k(u, u') = \mathrm{cov}(g(u), g(u'))$, and compactly define a covariance vector $\{\mathbf{k}_* \in \mathbb{C}^{D\times 1} | \mathbf{k}_*[i] = k(u_*, u_i)\}$ where each element is the covariance between the test point at $u_*$ (the point to be estimated) and the $D$ training points. The observations of each training point are contained in $\mathbf{v} = [v_1, v_2, ..., v_D]^T$. The training set covariance matrix $\{\mathbf{K} \in \mathbb{C}^{D\times D} | \mathbf{K}_{ij} = k(u_i, u_j)\}$ is constructed such that the test point estimate $v_*$ and estimate variance $\sigma^2_{v_*}$ are given as

$$v_* = \mathbf{k}_*^H \mathbf{K}_v^{-1} \mathbf{v} \tag{7}$$

$$\sigma^2_{v_*} = k(u_*, u_*) - \mathbf{k}_*^H \mathbf{K}_v^{-1} \mathbf{k}_*, \tag{8}$$

where $\mathbf{K}_v = \mathbf{K} + \sigma_n^2 \mathbf{I}_D$ and $\sigma_n^2$ is the channel noise as described in II-B. The uncertainty given in (8) gives the predictability needed later for model complexity determination.

The choice of the GPR kernel in relation to the statistics of sample data has a profound impact on the effectiveness of GPR learning. A previous study [14] used a radial basis function (RBF) for the GPR kernel, which is given by $k(u, u') = \theta_1 \exp(-(u - u')^2/\theta_2)$ where $\theta_1$ and $\theta_2$ are hyperparameters for the RBF kernel. $\theta_2$ is commonly referred to as the "length-scale" of the RBF, and controls the sharpness of the Gaussian-like distribution. Instead of using the same RBF kernel of GPR to cope with the fast time-varying mobile channels, we propose a new kernel, called a Bessel kernel, based on the application of Jakes' Gaussian process model [10], to capture a Bayesian prior for the GPR latent function (channel) that is aligned to a practical mobile medium.

### B. Bessel Kernel for Mobile Channels

We now observe the duality of the MMSE estimate in (5) and the GPR test point estimate in (7). Both take observations of Gaussian-distributed source and noise vectors, apply a filtering vector that comprises of correlation/covariance matrices, and produce *a posteriori* Gaussian density functions. If we form the training set covariance matrix $\mathbf{K}$ to take the form of the correlation matrix of channel estimates $\mathbf{R}_{hh}$, then (7) is in the form of (5) when $\mathbf{k}_*^H$ is the first row of $\mathbf{K}$, and $\mathbf{v}$ consists of the least-squares channel estimates $\hat{\mathbf{h}}^{LS}$; the solution to (7) results in the MMSE estimate for that pilot symbol corresponding to the first training point in $\mathbf{v}$. As autocorrelation is simply a normalized covariance for zero-mean data, we therefore propose, based on the time autocorrelation function in (3), a Bessel kernel for GPR channel estimation in Rayleigh fading channels as follows:

$$k(u, u') = \theta_1 J_0(2\pi\theta_2\tau), \tag{9}$$

where $\theta_1$ and $\theta_2$ are hyperparameters for the Bessel kernel, and $\tau = (u - u')T$.[2] The $\theta_2$ in (9) is essentially the Doppler frequency $f_m$, while $\theta_1$ effectively accounts for variations in received power $P$ defined in (3).

### C. Hyperparameter Training

Hyperparameters are trained to best fit the model $g(\cdot)$ to the observed training points $\mathbf{v}$. We accomplish this supervised learning by minimizing the negative log marginal likelihood of the observations, i.e., $\min_{\boldsymbol{\theta}} -\log p(\mathbf{v}|\mathbf{u}, \boldsymbol{\theta})$, where $\boldsymbol{\theta} = [\theta_1, \theta_2]^T$ is the hyperparameter vector for the kernel $k(\cdot)$. The criterion can be equivalently rewritten as

$$\min_{\boldsymbol{\theta}} \frac{1}{2}\left(\mathbf{v}^T \mathbf{K}_v^{-1} \mathbf{v} + \log(\det \mathbf{K}_v) + D\log(2\pi)\right). \tag{10}$$

Since optimization of (10) is a standard procedure in GPR training, we omit the details here. Any iterative gradient-based algorithm can be used to find the hyperparameters $\boldsymbol{\theta}$. These algorithms require computing the partial derivative of the objective in (10), which is given by [9]

$$-\frac{\partial \log p(\mathbf{v}|\mathbf{u}, \boldsymbol{\theta})}{\partial \theta_j} = -\frac{1}{2}\mathrm{tr}\left[(\alpha\alpha^T - \mathbf{K}_v^{-1})\frac{\partial \mathbf{K}_v}{\partial \theta_j}\right], \ \forall j, \tag{11}$$

where $\alpha = \mathbf{K}_v^{-1}\mathbf{v}$ and the partial derivative $\partial \mathbf{K}_v / \partial \theta_j$ is readily computed entry-wise. The gradients of the Bessel kernel in (9) are given by $\partial k / \partial \theta_1 = J_0(2\pi\theta_2\tau)$ and $\partial k/\partial\theta_2 = -2\pi\tau\theta_1 J_1(2\pi\theta_2\tau)$ where $J_1(\cdot)$ is the first order Bessel function of the first kind.[3]

### D. GPR for Interpolation of Channel Estimates

After the hyperparameters are learned, we can now use GPR to non-linearly interpolate information symbol channel estimates in between pilots without additional training. The trained kernel is reused to find the estimates, in conjunction with (7) and (8). To the best of our knowledge, DNNs have not yet been shown to do this using a single trained DNN for both pilot channel estimates and interpolated information symbol channel estimates.

To perform time-domain interpolation using GPR, we compute each pilot symbol's LS channel estimate in the $j$th slot as $\hat{h}^{LS}_{jL}$. Using current and previous values of $\hat{h}^{LS}$ as the training points for GPR, we can then regress the $L$-1 channel estimates across time in between the pilot symbols of slot $s$-1 and $s$ using the kernel function $k(\cdot)$ in (9), where slot $s$ is the current received slot. This concept is shown in Fig. 3.

We form the time-domain GPR training data set $\mathcal{S} = \{(Li, \hat{h}^{LS}_{s-i}), i = 0, ..., D-1\}$ to create symbol index and pilot estimate pairs. We define pilot channel estimate vector $\{\mathbf{w} \in \mathbb{C}^{D\times 1} | \mathbf{w} = [\hat{h}^{LS}_s, \hat{h}^{LS}_{s-1}, \ldots, \hat{h}^{LS}_{s-D+1}]^T\}$ as the last $D$ pilot symbol LS estimates to find the channel estimate $\hat{h}^{\mathrm{gp}}(l)$ corresponding to the $l$-th symbol of the previous slot, i.e., interpolation, where from (7) and (8),

$$\hat{h}^{\mathrm{gp}}(l) = \mathbf{k}_*^H[L-l](\mathbf{K} + \sigma_n^2\mathbf{I}_D)^{-1}\mathbf{w}, \tag{12}$$

$$\sigma^2_{\mathrm{gp}}(l) = k(0,0) - \mathbf{k}_*^H[L-l](\mathbf{K} + \sigma_n^2\mathbf{I}_D)^{-1}\mathbf{k}_*[L-l], \tag{13}$$

where $\mathbf{k}_*[l] = [k(0,l), k(L,l), \ldots, k(L(D-1),l)]^T$ and $\mathbf{K}(i,j) = k(Li, Lj)$, $\forall i,j$ with $k(\cdot)$ as defined in (9). (13)

[2] While we focus on Rayleigh fading models in this work, it is worthwhile to point out that our approach can be easily extended to other parameterized channel models such as Ricean and Nakagami fading models.

[3] Because the kernel described here is a covariance matrix comprised of linearly-spaced observations, $\mathbf{K}$ is Toeplitz, and inversion of $\mathbf{K}$ can be computed efficiently in $\mathcal{O}(4D^2)$ time per optimization iteration, but calculation of the derivatives is then at $\mathcal{O}(D^2)$ time.

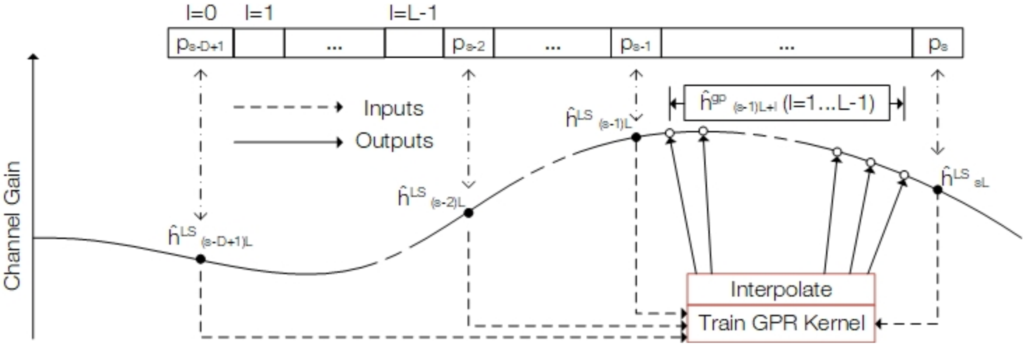

Fig. 3. Using pilot symbol channel estimates from $\hat{h}^{LS}$ to learn the channel and obtain information symbol channel estimates $\hat{h}^{gp}$

asymptotically approaches zero as $D$ increases; therefore $D$ is chosen to balance performance with the computational complexity inherent in the inversion of $\mathbf{K}$.

We now have channel estimates for information symbols, and can analytically predict BER performance given the channel estimate variance for each symbol position in the slot.

## V. Performance Analysis

In this section, we analyze and approximate BER and use it to find a practical bound on the number of training points needed to achieve near-optimal performance. We focus on quadrature phase shift keying (QPSK) for simplicity.

Beginning with the signal model introduced in (1) and the subsequent decomposition of the received signal into the original signal and amplitude/phase estimation error in (2), we arrive at the representation

$$y = (a+\delta a) + \frac{n}{\hat{h}} = a + \left(-a\frac{\delta h}{h+\delta h} + \frac{n}{h+\delta h}\right) = a + e. \quad (14)$$

Regarding the first equality in (14), the received signal $y$ consists of the original signal $a$ and the decoder error $\delta a$, plus the AWGN term scaled by the channel estimate $\hat{h}$. For the second equality in (14), we express the received signal in terms of the original signal $a$ and a composite error term $e$ comprised of two Gaussian-distributed random variables $\delta h \sim \mathcal{CN}(0, \sigma_{\text{gp}}^2)$ and $n \sim \mathcal{CN}(0, \sigma_n^2)$

Now, we attempt to linearize the error term $e$ in (14). For QPSK, let $\mathcal{E}_s$ be the signal energy and $\mathcal{E}_b = \mathcal{E}_s/2$ be the per-bit signal energy. For high SNR ($\sigma_n^2 \ll \mathcal{E}_s$) and small GP variance ($|\delta h| \ll |h|$),

$$\frac{\delta h}{h+\delta h} \approx \frac{\delta h}{h} \text{ and } \frac{n}{h+\delta h} \approx \frac{n}{h}. \quad (15)$$

Then (14) can be rewritten as

$$y \approx a + \frac{1}{h}(n - a\delta h) = \frac{a(h-\delta h)}{h} + \frac{n}{h}. \quad (16)$$

Note that (16), when $h$ is a fading term, is in the form of signal plus noise in fading. The BER for QPSK in Rayleigh fading (assuming the channel coherence time is sufficiently greater than the symbol time) is given by [15]

$$P_e^{\text{QPSK}} \approx \frac{1}{2}\left(1 - \sqrt{\frac{\mathcal{E}_b/\widetilde{N}_0}{1+(\mathcal{E}_b/\widetilde{N}_0)}}\right), \quad (17)$$

where $\widetilde{N}_0 = \sigma_n^2 + \mathcal{E}_s\sigma_{\text{gp}}^2$ is the sum of the noise power from the random variables $n$ and $a\delta h$ in (16). Since $\widetilde{N}_0$ is a function of $l$ as shown in (13) we average (17) across all symbol positions in the slot to arrive at the approximation for the QPSK bit error rate using GPR,

$$P_e^{\text{QPSK}}(L) \approx \frac{1}{2(L-1)}\sum_{l=1}^{L-1}\left(1 - \sqrt{\frac{\mathcal{E}_b/(\sigma_n^2 + \mathcal{E}_s\sigma_{\text{gp}}^2(l))}{1+(\mathcal{E}_b/(\sigma_n^2 + \mathcal{E}_s\sigma_{\text{gp}}^2(l)))}}\right). \quad (18)$$

### A. Determination of training data set size

Armed with the BER using GPR, we now seek to provide a practical bound on $D$, and can illustrate a complexity bound for GPR in receivers by recognizing that the noise term $\widetilde{N}_0$ can be dominated by either $\sigma_n^2$ or $\mathcal{E}_s\sigma_{\text{gp}}^2$. Recall in (13) that the GP variance is inversely proportional to $D$. If the GP variance is less than the observation noise, the observation noise will dominate the BER term in (18) and it can be surmised the number of training points $D$ can be reduced without appreciable loss of performance. There is then a value $D_{opt}$ as the lowest number of training points needed to have the GPR variance equal to or less than the noise power, such that neither noise term in (18) dominates $\widetilde{N}_0$.

We can see this effect in Fig. 4 for different mobile velocities using LTE parameters of $L$=6, $f_c$=2.0 GHz, and $T$=71.4$\mu s$, where the BER is high with few training points $D$ and $\sigma_{\text{gp}}^2 \gg \sigma_n^2$, but then decreases as $D$ increases until $\sigma_{\text{gp}}^2 \approx \sigma_n^2$, at which point any further increase in $D$ results in negligible decreases in BER since $\sigma_n^2$ now dominates the noise term. This occurs at approximately 4-6 training points. For reference, the irreducible QPSK BER for perfect channel estimates across all pilot and information symbols is shown.

Notably, as velocity increases, more of the off-diagonal elements of $\mathbf{K}$ tend to zero, increasing $\det \mathbf{K}$ and causing a higher variance in (13). At some point, the GPR variance across all information symbols exceeds the noise floor for any value of $D$. This can be seen for the curve at 600 kph.

## VI. Simulation Results

Fig. 5 shows the normalized MSE from a PSAM simulation using QPSK. LS channel estimates are plotted with GPR estimates using Bessel kernel, $D$=6, and training using

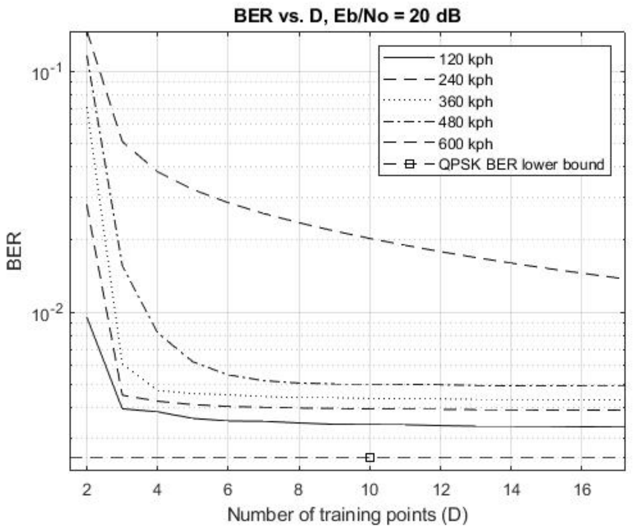

Fig. 4. BER vs. Number of training points for different mobile velocities

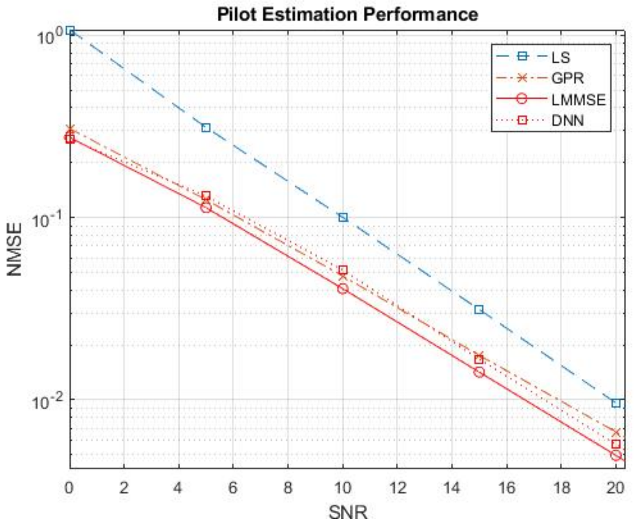

Fig. 5. Normalized MSE vs. SNR for GPR and DNN

stochastic gradient descent. Curves assume a pilot spacing of $L$=10, $T$=71.4$\mu s$, and Rayleigh fading with $f_mT$=0.016.

Channel estimates from the DNN as described in [6] are also overlaid on the plot. In the paper, the DNN is a fully-connected three-layer DNN with 512, 256, and 128 neurons using rectified linear unit (ReLU) activation functions trained with the Adam algorithm. The actual (noise-free) channel estimates are used to train the DNN, then the DNN is updated in real-time using LS (noisy) estimates.

It can be seen that the trained GPR performs as well as the DNN, and both are close to the theoretical LMMSE lower bound. For the simulated parameters, all machine learned-estimates improve LS estimates by approximately 4-6 dB.

It can thus be surmised that the hyperparameters of the GPR Bessel kernel can converge such that the kernel returns the channel autocorrelation, while the fully-connected DNN converges to act as a filter on the LS inputs similar to how the terms in (5) act to filter $\hat{\mathbf{h}}^{LS}$.

## VII. Conclusion

We introduced the concept of machine learning using GPR and applied it to PSAM for use in forming a proxy for $\mathbf{R}_{hh}$ to get the LMMSE channel estimates for pilot symbols, thus showing the interpretability of GPR. GPR also enables interpolating channel estimates between pilots when in a small-scale fading environment. We formulated the generalized expression for QPSK BER when using GPR on PSAM pilots in Rayleigh fading. The predictability of channel estimate MSE lets us predict the necessary model complexity to approach the BER floor for a given noise floor and pilot spacing $L$. Finally, the theoretical MSE for GPR was compared with simulation results for GPR-derived and DNN-derived channel estimates, and showed that performance between GPR and DNN are comparable, with both shown to be close to the MMSE bound.

## Acknowledgement

The authors appreciate the support of Anushka Bhattacharya at the University of Kansas for implementation of the DNN for simulation results.

## References

[1] 3GPP, "Service requirements for the 5G system; Stage 1," Technical Specification (TS) 22.261, 3rd Generation Partnership Project (3GPP), 2019. Version 17.0.0.

[2] 3GPP, "Requirements for further advancements for E-UTRA (LTE-Advanced)," Technical Report (TR) 36.913, 3rd Generation Partnership Project (3GPP), 2020. Version 16.0.0.

[3] J. K. Cavers, "An Analysis of Pilot Symbol Assisted Modulation for Rayleigh Fading Channels," *IEEE Trans. on Vehicular Technology*, vol. 40, pp. 686–693, Nov. 1991.

[4] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 563–575, 2017.

[5] H. Ye, G. Le, and B.-H. Juang, "Power of deep learning for channel estimation and signal detection in OFDM systems," *IEEE Wireless Communications Letters*, vol. 7, no. 1, pp. 114–117, 2018.

[6] X. Ma, H. Ye, and Y. Li, "Learning assisted estimation for time-varying channels," in *2018 15th International Symposium on Wireless Communication Systems (ISWCS)*, pp. 1–5, 2018.

[7] Q. Hu, F. Gao, H. Zhang, S. Jin, and G. Y. Li, "Deep learning for channel estimation: Interpretation, performance, and comparison," *IEEE Transactions on Wireless Communications*, vol. 20, no. 4, pp. 2398–2412, 2021.

[8] H. Huang, S. Guo, G. Gui, Z. Yang, J. Zhang, H. Sari, and F. Adachi, "Deep learning for physical-layer 5G wireless techniques: Opportunities, challenges and solutions," *IEEE Wireless Communications*, vol. 27, no. 1, pp. 214–222, 2020.

[9] C. E. Rasmussen and C. K. I. Williamst, *Gaussian Processes for Machine Learning*. Cambridge, Mass.: The MIT Press, 2006.

[10] W. C. Jakes, *Microwave Mobile Communications*. New York: Wiley, 1974.

[11] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA); User Equipment (UE) radio transmission and reception," Technical Specification (TS) 36.101, 3rd Generation Partnership Project (3GPP), 2012. Version 10.6.0.

[12] T. S. Rappaport, *Wireless Communications*. New Jersey: Prentice-Hall, 1996.

[13] Y. Liu, Z. Tan, H. Hu, L. J. Cimini, and G. Y. Li, "Channel estimation for OFDM," *IEEE Communications Surveys Tutorials*, vol. 16, no. 4, pp. 1891–1908, 2014.

[14] F. Pérez-Cruz, J. J. Murillo-Fuentes, and S. Caro, "Nonlinear Channel Equalization With Gaussian Processes for Regression," *IEEE Trans. on Signal Processing*, vol. 56, pp. 5283–5286, Oct. 2008.

[15] J. G. Proakis, *Digital Communications*. McGraw-Hill, 3rd ed., 1995.