Modeling of the tire-road friction using neural networks including quantification of the prediction uncertainty

Magnus Malmström, Isaac Skog, Daniel Axehill and Fredrik Gustafsson

Malmström, M., Skog, I., Axehill, D., Gustafsson, F., (2021), Modeling of the tire-road friction using neural networks including quantification of the prediction uncertainty, *2021 IEEE 24th International Conference on Information Fusion (FUSION)*. https://doi.org/10.23919/fusion49465.2021.9626974

Original publication available at:
https://doi.org/10.23919/fusion49465.2021.9626974

Modeling of the tire-road friction using neural networks including quantification of the prediction uncertainty

Magnus Malmström
Linköping University
magnus.malmstrom@liu.se

Isaac Skog
Linköping University
isaac.skog@liu.se

Daniel Axehill
Linköping University
daniel.axehill@liu.se

Fredrik Gustafsson
Linköping University
fredrik.gustafsson@liu.se

***Abstract*—Despite the great success of neural networks (NN) in many application areas, it is still not obvious how to integrate an NN in a sensor fusion framework. The reason is that the computation of the for fusion required variance of NN is still a rather immature area. Here, we apply a methodology from system identification where uncertainty of the parameters in the NN are first estimated in the training phase, and then this uncertainty is propagated to the output in the prediction phase. This local approach is based on linearization, and it implicitly assumes a good signal-to-noise ratio and persistency of excitation. We illustrate the proposed method on a fundamental problem in advanced driver assistance systems (ADAS), namely to estimate the tire-road friction. This is a single input single output static nonlinear relation that is simple enough to provide insight and it enables comparisons with other parametric approaches. We compare both to existing methods for how to assess uncertainty in NN and standard methods for this problem, and evaluate on real data. The goal is not to improve on simpler methods for this particular application, but rather to validate that our method is on par with simpler model structures, where output variance is immediately provided.**

I. Introduction

Vehicles today have an increasing level of autonomy, many of which rely on the progress of computer vision algorithms for situational awareness. Examples of ADAS that rely on computer vision include lane keeping assistance, intelligent speed control and collision avoidance systems. The latter one is enabled by the impressive performance of NN to classify different road objects such as other vehicles, humans, bicycles, animals and other hazards. Though objects are mostly correctly classified, commercial autonomous cars have also suffered from spectacular failures, several leading to fatal accidents [1], [2]. In many of these cases, the radar has had the ability to detect the obstacle, but the system relied on the NN assessment based on vision sensors. If all obstacle detection systems could provide reliable uncertainty to its output, then proper sensor fusion could be applied.

We do not attempt to attach the full ADAS challenge in this contribution, but rather start on one simple but fundamental sub-problem, namely to estimate the tire-road friction. Knowledge of the friction level is crucial for which manoeuvres that can be applied and the minimum braking distance. It is of utmost importance that an ADAS do not overestimate the friction, while it is also important to not be too conservative if the friction is good. If an NN eventually is to be used in an end-to-end ADAS design, uncertainty assessment must be solved successfully internally in the NN, which uncertainty the friction estimation is one component of.

A method to compute output variance from an NN was proposed in [3]. It relies on well-established methodology in system identification [4]. During the training phase, the covariance of the parameters (i.e. weights in the NN) are estimated using a local approach. This covariance can be computed with little extra computational cost by utilizing the gradient from a back-propagation algorithm. In the prediction phase, the covariance of the parameters can be transformed to a variance of the output, again using a local approach based on linearization of the NN.

The tire-road friction curve is usually modeled as a static nonlinear mapping from one input (wheel slip) to one output (maximum traction force) [5], [6]. The simplicity of the model makes it possible to visualize the computed uncertainty, and compare to the many standard methods that exist for this particular application. Our ambition is not to compete with these tailored methods, or the other methods we compare with, but rather to validate that the local approach provides a variance of the friction which is as reliable as other methods; [7], [8], [9], [10], [11]. This makes it plausible that NNs can eventually be developed for autonomous braking and steering ADAS.

II. Problem formulation

Consider the following nonlinear static model

$$y_m = s_m + e_m \tag{1a}$$
$$s_m = f^*(\mathbf{x}_m) \tag{1b}$$

where $y_m \in \mathbb{R}^{d_s}$ is the observation of the output $s_m \in \mathbb{R}^{d_s}$, $\mathbf{x}_m \in \mathbb{R}^{d_x}$ is the input, and the function $f^*(\mathbf{x}_m)$ describes the relation between the input and the output of the *true system* $\mathcal{S}$ under consideration at time index m. Furthermore, e_m is the observation noise which is identically and independently distributed according to some distribution with variance λ_0 and mean $E[e_m] = 0$. Given a set of data consisting of the observations $y_{1:N} \triangleq (y_n)_{n=1}^N$ and inputs $\mathbf{x}_{1:N} \triangleq (\mathbf{x}_n)_{n=1}^N$,

This work is supported by Sweden's innovation agency, Vinnova, through project iQDeep (project number 2018-02700).

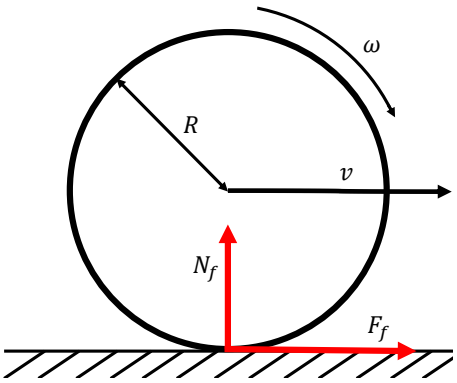


Fig. 1: Single wheel dynamics.

a parametric model in some model set, $f(\mathbf{x}_m, \boldsymbol{\theta}) \in \mathcal{M}^*$, with parameter vector $\boldsymbol{\theta} \in \Theta \subset \mathbb{R}^d$ containing d parameters is fitted to the data. This can be done by e.g., minimizing a least-squares cost function between the model and the data. Then, at new time instances a prediction of the signal, $\hat{s}_m = f(\mathbf{x}_m, \hat{\boldsymbol{\theta}}_N)$ can be created. Here, the parameters $\hat{\boldsymbol{\theta}}_N$ are estimated[1] as,

$$\hat{\boldsymbol{\theta}}_N = \arg\min_{\boldsymbol{\theta}} V_N(\boldsymbol{\theta}), \tag{2a}$$

$$V_N(\boldsymbol{\theta}) \triangleq \frac{1}{N}\sum_{n=1}^{N} ||y_n - f(\mathbf{x}_n, \boldsymbol{\theta})||^2. \tag{2b}$$

The aim of this paper is to quantify the uncertainty in the prediction for parametric models, and to compare the expression for different methods for quantifying the uncertainty as well as the use of different models. This is equivalent to comparing the prediction error variance, where prediction error is given as $\varepsilon(\mathbf{x}_{1:N}, \hat{\boldsymbol{\theta}}_N) \triangleq y_{1:N} - \hat{s}_{1:N}$. Through out the analysis, it is assumed that the model set includes the true system, $\mathcal{S} \in \mathcal{M}^*$.

The true system under consideration describes how the normalized traction force (NTF) depends on the wheel slip of a car, which is a central component in ADAS. Here slip, $s = (\omega R - v)/v$, and NTF, $\mu = F_f/N_f$, are computed using the angular velocity of the driven wheel ω, the wheel radius R, the velocity of the vehicle v, the traction force between the tire and the road F_f, and the normal force N_f. This can be derived from the single wheel dynamics seen in Fig. 1. In practice, F_f and N_f are not directly measurable, and need to be estimated by using data from sensors measuring other quantities, for details see [12].

III. Linearization method to quantify uncertainty in prediction for neural networks

The linearization method is a commonly used method in system identification and statistics to quantify the uncertainty in the prediction from a model [4], [13], but not a standard for NNs. One reason for this is a consequence of the ambiguities in the parameterization of NNs that are required to be taken in consideration for the linearization method.

[1]Also referred to as learning the parameters.

A. Neural networks

One example of nonlinear black-box models are NNs [14]. They can be described by the recursion

$$\mathbf{h}^{(0)} = \mathbf{x}_m, \tag{3a}$$

$$\mathbf{a}^{(l+1)} = \begin{pmatrix}\mathbf{h}^{(l)} & 1\end{pmatrix}^\top W^{(l)}, \quad l = 0, \cdots, L, \tag{3b}$$

$$\mathbf{h}^{(l)} = \sigma\big(\mathbf{a}^{(l)}\big), \quad l = 1, \cdots, L, \tag{3c}$$

$$s_m = a^{(L)}, \tag{3d}$$

where a nonlinear transformation, $\sigma(.)$ is done in every layer. The nonlinearity is called activation function, commonly used choices of activation functions are e.g., the sigmoid function, the hyperbolic tangent, the rectified linear unit (ReLU).

Here, the number of layers in the network is denoted L, the weights of the lth layer $W^{(l)}$, the contribution from layer $(l-1)$ to layer l $\mathbf{a}^{(l)}$, and the activation of the contribution from the $(l-1)$th layer at the lth layer $\mathbf{h}^{(l)}$, which are referred to as the *hidden node*. Collecting the biases and the weights of the NN into a parameter vector, i.e.,

$$\boldsymbol{\theta} = \begin{pmatrix}\text{vec}(W^{(0)}) & \cdots & \text{vec}(W^{(L)})\end{pmatrix}^\top, \tag{4}$$

an NN can be written as a parametric model, $\hat{s}_m = f(\mathbf{x}_m, \boldsymbol{\theta})$. The notation $[d_x, d_1, \cdots, d_L, d_s]$ is used, where d_x is the number of inputs, d_s is the number of outputs and d_l is the number of nodes in lth hidden layer.

B. Linearization method

By solving (2a), the parameters $\hat{\boldsymbol{\theta}}_N$ is found, which is used to create an estimate of the signal $\hat{s}_m$. Given knowledge about the covariance of the parameters $\text{Cov}(\hat{\boldsymbol{\theta}}_N)$, the prediction error variance of the signal can be calculated as

$$\text{Var}(\hat{s}_m) = f'_{\boldsymbol{\theta}}(\mathbf{x}_n, \hat{\boldsymbol{\theta}}_N)\text{Cov}(\hat{\boldsymbol{\theta}}_N)f'_{\boldsymbol{\theta}}(\mathbf{x}_n, \hat{\boldsymbol{\theta}}_N)^\top, \tag{5}$$

i.e., the prediction error variance is found by linearizing the model, where $f'_{\boldsymbol{\theta}}(\mathbf{x}_n, \hat{\boldsymbol{\theta}}_N) \triangleq \frac{\partial}{\partial \boldsymbol{\theta}} f(\mathbf{x}_n, \boldsymbol{\theta})\big|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}_N}$. In [3], it is shown that given that the model set for the choose model is flexible enough to include the true system, the parameter covariance is computed as

$$\text{Cov}(\hat{\boldsymbol{\theta}}_N) \sim \frac{\hat{\lambda}_N}{N}\mathcal{I}_{\boldsymbol{\theta}}^{+}, \quad \hat{\lambda}_N = \frac{1}{N}\sum_{n=1}^{N}\varepsilon^2(\mathbf{x}_n, \hat{\boldsymbol{\theta}}_N), \tag{6a}$$

$$\mathcal{I}_{\boldsymbol{\theta}} = \frac{1}{N}\sum_{n=1}^{N}(f'_{\boldsymbol{\theta}}(\mathbf{x}_n, \hat{\boldsymbol{\theta}}_N))^\top f'_{\boldsymbol{\theta}}(\mathbf{x}_n, \hat{\boldsymbol{\theta}}_N). \tag{6b}$$

This method is valid for any parametric model, in particular if the model is linear in its parameters, such as Laplacian polynomials. For NNs, due to ambiguities in the model structure of NN, the parameter covariance might be singular (or close to singular). Previously, the linearization method have been applied to NNs under the assumption of minimal representation and identifiability [15], [16], [17], [18], [19], and in [3], the method has been extended to handle ambiguities stemming from the model structure of NNs as well as handle the case when an overparameterized NN is used. Hence, even though it might exist ambiguities in the model structure of the NN,

the linearization method can efficiently be used to quantify the uncertainty in the prediction for NNs.

IV. Related methods to quantify uncertainty

The subject of assessing uncertainty in the prediction of NNs have recently received increased attention [20], [10], [21], [22], [23]. Two of the most commonly used methods in the literature to quantify uncertainty in the prediction of NNs are; an ensemble technique, and an NN that learns its own uncertainty.

A. Ensemble techniques

A straightforward approach to quantify the uncertainty in the prediction from a model is to generate multiple realizations and calculate the variance of the ensemble of realizations. To create these realizations, either the data used to create the models should be perturbed by, e.g., adding noise to the observations, or changing the split between training data and validation data, or the parameter initialization should be different between the different realizations. For NN, the idea of combining models has also been shown to improve the prediction performance [24], [25].

This method is however not feasible for NNs, since already training a network using a single realization can be very time-consuming. Therefore, clever ways to create these ensembles have been proposed, e.g., using regularization techniques that are already used in training of NNs to prevent overfitting. Two such methods are Monte Carlo (MC) dropout and MC batchnorm, [26], [8].

In system identification for nonlinear models, it is also difficult to obtain uncertainty bounds, and if there is a structural model error, one must rather rely on variability obtained from repeating the experiment with varying excitation, i.e., ensembles [27]. Hence for any suggested method for quantifying the uncertainty in the prediction, it is essential to compare the results to the variance of an ensemble of models.

B. Learning prediction uncertainties in neural networks

Another method to obtain the prediction error variance from a black-box model is to include the variance as an output of the model [11], [10], [9]. Then, the loss function has to be modified, as well as some assumptions of the distribution of the prediction is required.

Assuming that the likelihood of the data is Gaussian, let the NN learn the logarithm of the variance and the mean, i.e.,

$$p(y_m|\mathbf{x}_m;\boldsymbol{\theta}) = \mathcal{N}(y_m;[f(\boldsymbol{\theta},\mathbf{x}_m)]_1,\exp([f(\boldsymbol{\theta},\mathbf{x}_m)]_2)), \quad (7a)$$

$$f(\boldsymbol{\theta},\mathbf{x}_m) = [\mu_y,\log(\sigma_y^2)]^\top, \quad (7b)$$

where μ_y and σ_y^2 is the mean and the variance of the signal y_m at time instance m, respectively. Here, the notion $[f(\boldsymbol{\theta},\mathbf{x}_m)]_i$ is used for the ith element of $f(\boldsymbol{\theta},\mathbf{x}_m)$. The maximum likelihood estimate (MLE) could then be found by minimizing the cost function

$$V_N(\boldsymbol{\theta}) = \frac{1}{N}\sum_{n=1}^{N}\frac{(y_n - [f(\boldsymbol{\theta},\mathbf{x}_n)]_1)^2}{\exp([f(\boldsymbol{\theta},\mathbf{x}_n)]_2)} + [f(\boldsymbol{\theta},\mathbf{x}_m n)]_2. \quad (8)$$

The model described in (7b) is referred to as the *Gaussian model* [10], [9].

V. Modeling normalized traction force

In this paper, comparison is also made between NNs and models from other model sets. The choices of models include both black-box models, i.e., universal approximators, as well as gray-box models, were physical insights have been used to create the model.

A. Magic tire formula

Extensive modeling of the tire-road friction has resulted in the gray-box model called *magic tire formula*, [5], given by

$$\mu = D\sin\big(C\arctan((1-E)s + \frac{E}{B}\arctan(Bs))\big), \quad (9)$$

where the parameters B, C, D, E depend on the surface. In this paper, a bias term is added to the model, i.e., removing the assumption that zero slip does not necessarily result in zero NTF. This is necessary since the observed data is quite noisy.

B. Laplacian polynomials

When estimating a model that describes the relationship between the input $\mathbf{x}_m$ and the output s_m, it is a good practice to start by restricting the model set to simple models, then complexity can be added to the model. This is of particular importance when the model set consists of black-box models, i.e., universal approximators of functions. For example, models that are linear in the parameters, i.e., $f(\mathbf{x}_m,\boldsymbol{\theta}) = \boldsymbol{\varphi}^\top(\mathbf{x}_m)\boldsymbol{\theta}$, are good choices for simpler models to start with. Here $\boldsymbol{\varphi}^\top(\mathbf{x}_m)$ is called the regressors (or basis function), which is some transformation of the input $\mathbf{x}_m$. The result from more complex models can then later be compared against these simpler models, this to validate the use of the more complex models.

A choice of model linear in the parameters with orthogonal basis functions is the Laplacian polynomials [31], which are given by

$$[\boldsymbol{\varphi}^\top(\mathbf{x}_m)]_j = \sin\big(\pi j(\mathbf{x}_m + L)/(2L)\big)/\sqrt{L}, \quad (10)$$

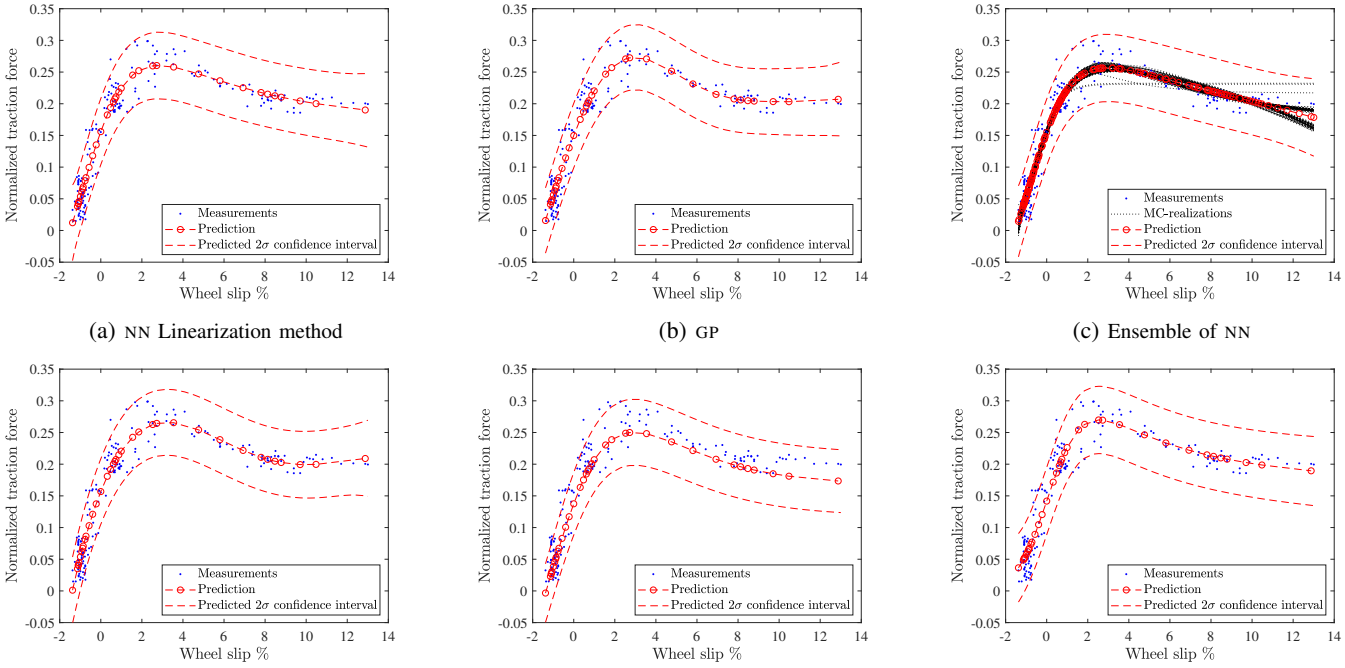
where L is a hyperparameter set by the user.

C. Gaussian processes

An example of a non-parametric model is Gaussian processes (GPs), [32]. The concept of uncertainty is naturally included in GPs, where the kernel function represents some covariance function for the learned function. A GP is a collection of random variables that have a joint Gaussian distribution. It can be described completely by a mean function and a kernel function. Assuming $f(\mathbf{x}_m)$ to be some process, it can be approximated with a GP as $f(\mathbf{x}_m) \sim \mathcal{GP}(m(\mathbf{x}_m), k(\mathbf{x}_m,\mathbf{x}'))$, where $m(\mathbf{x}_m)$ is the mean of the real process, $k(\mathbf{x}_m,\mathbf{x}')$ is the covariance (called kernel function), and $\mathbf{x}'$ is data from $\mathbf{x}_{1:N}$ which is data used to train the GP. This interpretation of

TABLE I: Settings for the different models used to estimate the friction data. Including: implementation specifications such as chosen optimization algorithm, which model order (or model setting) that is used, and classification of the model into linear black-box (LBB) models, nonlinear black-box (NLBB) models, and nonlinear gray-box (NLGB) models.

Model	Learning algorithm	Model order	Classification
Laplacian polynomials, Section V-B	linear least squares solution	4 basis functions	LBB
Fully-connected NN, Section III-A	ADAM [28]	sigmoid activation function, $[1, 2, 1]$	NLBB
NN with learned uncertainty, Section IV-B	RMS-prop [29]	sigmoid activation function, $[1, 6, 6, 6, 2]$	NLBB
Magic tire formula with bias, Section V-A	Levenberg-Marquardt algorithm [30]	See (9) with added bias	NLGB
GP, Kernal: (11), Section V-C	MATLABS SML toolbox	$L = \frac{1}{N}\sum_{i=1}^{d_x}\sqrt{\mathrm{Var}[\mathbf{x}_m]_i}$, $\sigma_f^2 = \frac{1}{\sqrt{2}}\sqrt{\mathrm{Var}(y)}$	NLBB



(a) NN Linearization method (b) GP (c) Ensemble of NN

(d) Laplacian polynomials (e) NN with learned uncertainty (f) Magic tire formula with bias

Fig. 2: Prediction with quantification of the uncertainty of the prediction for different models describing the relationship between the NTF and the wheels slip. For NNs three different methods of quantifying the uncertainty are presented.

a GP is sometimes referred to as the *function-space view*. One example of such kernel is the squared exponential[2]

$$k(\mathbf{x}_m, \mathbf{x}') = \sigma_f^2 \exp\big(-(\mathbf{x}_m - \mathbf{x}')^2/(2L^2)\big), \qquad (11)$$

where L is the length scale, and σ_f^2 the signal variance.

VI. EXPERIMENTS

To compare the performance of the different models, the commonly used numeric measure model fit is used [4]. Under the assumption that the prediction error is Gaussian distributed, a 95% confidence interval is calculated for all the models, i.e., a 2σ interval where $\sigma_n = \sqrt{\mathrm{Var}(\hat{y}_n)}$, i.e., the variance of the estimated signal. This variance is calculated as the sum of the variance of the model and the variance of the measurement noise, i.e.,

$$\mathrm{Var}(\hat{y}_n) = \mathrm{Var}(\hat{s}_n) + \lambda_N. \qquad (12)$$

The five different models used to approximate the relationship between the wheel slip and NTF are:

1) Laplacian polynomials described in Section V-B, which are a black-box model linear in the parameters with orthogonal basis functions.
2) Fully-connected NN describe in Section III-A with sigmoid as activation function.
3) NN with learned uncertainty described in Section IV-B, where there is assumption on Gaussian distribution on the prediction.
4) The magic tire formula with an added bias term described in Section V-A.
5) GP described in Section V-C with the squared exponential kernel.

As well as an ensemble of NNs as described in Section IV-A, which the mean of the ensemble is used as the prediction.

The data in the paper comes from tire-road friction experiments on snow conducted by NIRA Dynamics. Neither the wheel slip nor the NTF is measured but instead calculated using measurements of the wheel speed, acceleration of the vehicle, engine speed, and engine torque, for details see [12].

For the parametric models, the uncertainty in the prediction

[2]Sometimes called Radial Basis Functions or Gaussian.

TABLE II: Model fit on validation data for different models used to represent the relationship between the NTF and wheel slip.

Model	Fit to validation data
Laplacian polynomials, Section V-B	68.87
Fully-connected NN, Section III-A	67.04
NN with learned uncertainty, Section IV-B	63.30
Ensemble of NNs, Section IV-A	68.10
Magic tire formula with bias, Section V-A	68.26
GP, Section V-C	68.03

is calculated using the linearization method described in Section III-B , while for the Bayesian models the uncertainty is obtained by sampling from the posterior. For the different models, the model order is chosen such as the true system is included in the model set. In Table I the selected model order, classification if the models are linear or nonlinear, black-box or gray-box, and how the model is fitted to the data, are presented. The fit to the data for the different models and the ensemble of NNs are seen in Table II. All the model have similar fit. This is also seen in Fig. 2, where the prediction and prediction error variance for the different models are plotted. Since the fit is similar to the gray-box model magic tire formula, this would indicate that all models are suitable to model the relationship between the NTF and the wheel slip. The similarity between the prediction error variance also highlight the close connection between GPs, NNs, and Laplacian polynomials, [33], [34]. This indicates the usefulness of the linearization method to quantify the uncertainty in the prediction, since it gives similar results as to a non-parametric model, GPs, which natural includes a measure of uncertainty in the prediction.

Studying the prediction together with the prediction error variance seen in Fig. 2a, Fig. 2c, and Fig. 2e, the results look similar for the different methods to quantify the uncertainty in the prediction presented. If instead only the prediction error variance is studied, which is seen in Fig. 3, the linearization method and ensemble approach produce similar result, while the NN that learns its own uncertainty has problems to capture the uncertainty in the model. This might be a consequence of the high noise level of the data, where the NN learns the variance of the noise rather than the variance of the model.

VII. Conclusion

In search for a way to quantify the uncertainty in the predictions produced by NN, the application of methods for uncertainty estimation in system identification tools, has been investigated. More precisely, the feasibility and challenges of estimating the prediction error variance of the NN by linearizing the model and then propagating the uncertainty of the learned model parameters to the output have been investigated. The results show that, compared to existing methods based upon creating ensembles of NNs or extending the NN structure to also learn the uncertainty, the linearization method produces similar results for the estimated prediction error variance.

The result also shows that NNs are suitable to model the tire-road friction problem, this since they show similar results for both the prediction, compared to a collection of other models. The usefulness of the linearization method to quantified of

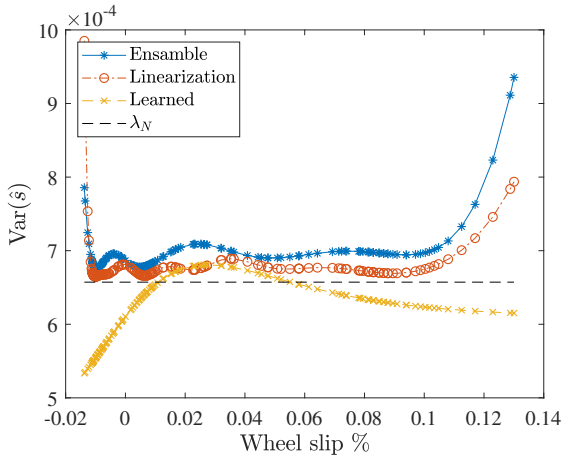


Fig. 3: Prediction error variance for NNs using ensemble of models, the linearization method, and an NN with learned uncertainty.

uncertainty in the prediction of NNs are further motivated by the similarity of the prediction error variance, compared to other models.

A challenge with the linearization method is that for a model with a large number of parameters, the required inversion to calculate the parameter covariance can be very computationally costly. This is something that will be considered in our future work.

References

[1] NTSB, "Preliminary Report Highway HWY18MH010," National Transportation Safty Board (NTSB), Technical Specification (TS), 05 2018. [Online]. Available: https://www.ntsb.gov/investigations/AccidentReports/Reports/HWY18MH010-prelim.pdf

[2] BBC News, "Tesla in fatal California crash was on Autopilot," 31 March 2018, [Last accessed: 9 Sep. 2020]. [Online]. Available: https://www.bbc.com/news/world-us-canada-43604440

[3] M. Malmström, "Uncertainties in neural networks a system identification approach," Licentiate Thesis, Dept. Elect. Eng., Linköping University, Linköping, Sweden, Apr. 2021.

[4] L. Ljung, *System identification: theory for the user (2nd edition)*. PTR Prentice Hall: Upper Saddle River, NJ, USA, 1999.

[5] H. Pacejka and I. Besselink, "Magic formula tyre model with transient properties," *Veh. syst. dynamics-Int. J. of Veh. Mechanics and Mobility*, vol. 27, no. S1, pp. 234–249, 1997.

[6] K. Berntorp and K. Hiroaki, "Bayesian learning of tire friction with automotive-grade sensors by gaussian-process state-space models," in *Proc. of IEEE 58th Conf. on Decision and Contr. , (CDC)*. Nice France: IEEE, 2019, pp. 6681–6686, 11-19 Dec.

[7] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning," in *Proc. of the 33td Int. Conf. on Mach. Learn. (ICML).*, New York, NY, USA, 2016, pp. 1050–1059, 20–22 Jun.

[8] M. Teye, H. Azizpour, and K. Smith, "Bayesian Uncertainty Estimation for Batch Normalized Deep Networks." in *Proc. of the 35th Int. Conf. on Mach. Learn. (ICML).*, 6–11 Jul 2018, Stockholm, Sweden,6–11 Jul.

[9] F. K. Gustafsson, M. Danelljan, G. Bhat, and T. B. Schön, "Energy-Based Models for Deep Probabilistic Regression," in *Proc. of 16th European Conf. on Comput. Vision (ECCV)*, Online, 2019, 23-28 Aug.

[10] A. Kendall and Y. Gal, "What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?" in *Adv. in Neural Inf. Process. Syst. (NIPS) 31*. Curran Associates, Inc., 4–9 Dec 2017, pp. 5574–5584, Long Beach, CA, USA, 4–9 Dec.

[11] A. Eldesokey, M. Felsberg, and F. S. Khan, "Propagating confidences through cnns for sparse data regression," in *British Mach. Vision Conf. (BMVC)*, 2018, p. 14, newcastle, UK, Sep 3-6.

[12] F. Gustafsson, "Slip-based tire-road friction estimation," *Automatica*, vol. 33, no. 6, pp. 1087–1099, 1997.

[13] H. Liero and S. Zwanzig, "Introduction to the theory of statistical inference." in *Chapman and Hall CRC Texts in Statistical Science, Boca Raton, FL, USA*, 2011.

[14] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT Press, 2016, London, England.

[15] J. T. G. Hwang and A. A. Ding, "Prediction Intervals for Artificial Neural Networks," *J. Am. Stat. Assoc. (JSTOR)*, vol. 92, no. 438, pp. 748–757, 1997.

[16] I. Rivals and L. Personnaz, "Construction of confidence intervals for neural networks based on least squares estimation," *Elsevier J. Neural Netw.*, vol. 13, 2000.

[17] S. He and J. Li, "Confidence Intervals for Neural Networks and Applications to Modeling Engineering Materials," in *Artificial Neural Netw.* IntechOpen, 2011, ch. 16.

[18] G. Papadopoulos, P. Edwards, and A. Murray, "Confidence estimation methods for neural networks: a practical comparison," *IEEE Trans. Neural Netw.*, 2001.

[19] G. Chryssoloiuris, M. Lee, and A. Ramsey, "Confidence interval prediction for neural network models," *IEEE Trans. Neural Netw.*, 1996.

[20] Z. Ghahramani, "Probabilistic machine learning and artificial intelligence." *Nature*, vol. 521, no. 7553, pp. 452 – 459, 2015.

[21] M. Garnelo, J. Schwatz, D. Rosenbaum, F. Viola, D. Rezende, S. M. A. Eslami, and Y. W. Teh, "Neural Processes," in *Proc. of the 35th Int. Conf. on Mach. Learn. (ICML) Workshop on Theo. Founda. and Appl. of Deep Generative Models*, Stockholm, Sweden, 2018, 10–15 Jul. [Online]. Available: https://arxiv.org/abs/1807.01622

[22] A. Kendall and R. Cipolla, "Modelling uncertainty in deep learning for camera relocalization," in *Proc. of IEEE Int. Conf. on Robot. and Autom. (ICRA)*, Stockholm, Sweden, 2016, pp. 4762–4769, 16–21, May.

[23] A. D'Amour, K. Heller, D. Moldovan, B. Adlam, B. Alipanahi, A. Beutel, C. Chen, J. Deaton, J. Eisenstein, M. D. Hoffman, F. Hormozdiari, N. Houlsby, S. Hou, G. Jerfel, A. Karthikesalingam, M. Lucic, Y. Ma, C. McLean, D. Mincu, A. Mitani, A. Montanari, Z. Nado, V. Natarajan, C. Nielson, T. F. Osborne, R. Raman, K. Ramasamy, R. Sayres, J. Schrouff, M. Seneviratne, S. Sequeira, H. Suresh, V. Veitch, M. Vladymyrov, X. Wang, K. Webster, S. Yadlowsky, Y. Taedong, X. Zhai, and D. Sculley, "Underspecification presents challenges for credibility in modern machine learning," *arXiv preprint arXiv:2011.03395*, 2020.

[24] J. G. Carney, P. Cunningham, and U. Bhagwan, "Confidence and prediction intervals for neural network ensembles," in *Proc. of the 6th Int. Joint Conf. Neural Netw. (IJCNN)*, Washington DC, DC, USA, 1999, 10-16, July.

[25] E. Ilg, O. Cicek, S. Galesso, A. Klein, O. Makansi, F. Hutter, and T. Brox, "Uncertainty estimates and multi-hypotheses networks for optical flow," in *Proc. of 15th European Conf. on Comput. Vision (ECCV)*, Munich, Germany, 2018, 8-14 Sep.

[26] Y. Gal, "Uncertainty in deep learning," Ph.D. dissertation, University of Cambridge, Jan. 2017.

[27] J. Schoukens and L. Ljung, "Nonlinear system identification: A user-oriented road map," *IEEE Control Syst. Mag.*, vol. 39, no. 6, pp. 28–99, 2019.

[28] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization." in *Proc. of 3rd Int. Conf. for Learn. Representations (ICLR)*, 7–9 May 2015, San Diego, CA, USA.

[29] T. Tieleman and G. Hinton, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," *COURSERA: Neural networks for machine learning*, 2012.

[30] J. Nocedal and S. Wright, *Numerical optimization*. Springer Science & Business Media,: New York, NY, USA, 2006.

[31] A. Solin and S. Särkkä, "Hilbert space methods for reduced-rank Gaussian process regression," *Statistics and Computing*, vol. 30, pp. 1–28, 2019.

[32] C. K. Williams and C. E. Rasmussen, *Gaussian processes for machine learning*. London, UK: MIT press Cambridge, MA, 2006, vol. 2, no. 3.

[33] R. M. Neal, *Bayesian learning for neural networks*. Springer Science & Business Media: New York, NY, USA, 1996, vol. 118.

[34] A. Svensson and T. B. Schön, "A flexible state–space model for learning nonlinear dynamical systems," *Automatica*, vol. 80, pp. 189–199, 2017.