# A Data-driven Generative Model for GPS Sensors for Autonomous Driving

Erik Karlsson
ÅF Technology AB
Gothenburg, Sweden
erik.a.karlsson@afconsult.com

Nasser Mohammadiha
Zenuity AB
Gothenburg, Sweden
nasser.mohammadiha@zenuity.com

## ABSTRACT

Autonomous driving (AD) is envisioned to have a significant impact on people's life regarding safety and comfort. Positioning is one of the key challenges in realizing AD, where global navigation systems (GNSS) is traditionally used as an important source of information. The area of GNSS are well explored and the different sources of error are deeply investigated. However the existing modeling methods often have very comprehensive requirements for the training data where all affecting conditions such as ephemeris data should be well known. The main goal of this paper is to develop a solution to model GPS error that only requires information which is available in the vehicle without having access to detailed information about the conditions. We propose a statistical generative model using autoregression and Gaussian mixture models and develop a learning algorithm to estimate the parameters using the data collected in real traffic. The proposed model is evaluated by comparing the produced artificial data with the validation data collected at different traffic conditions and the results indicate that the model is successfully mimicking the sensor behavior.

## CCS CONCEPTS

• **Computing methodologies** → **Modeling methodologies**; *Model verification and validation*;

## KEYWORDS

GPS modeling, Time series modeling, Autonomous driving

**ACM Reference Format:**
Erik Karlsson and Nasser Mohammadiha. 2018. A Data-driven Generative Model for GPS Sensors for Autonomous Driving. In *SEFAIAS'18: SEFAIAS'18:IEEE/ACM 1st International Workshop on Software Engineering for AI in Autonomous Systems , May 28, 2018, Gothenburg, Sweden,* Erik Karlsson (Ed.). ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3194085.3194089

## 1 INTRODUCTION

Self driving cars are among the technologies with a predicted high impact on our everyday life. Many sensors are usually installed on these cars to create an accurate 360 degrees perception of the surrounding of the car. Accurate positioning is one of the important components for the autonomous cars to safely navigate to the desired destination. Understanding and modelling the positioning errors is therefore very important for both optimal perception and navigation design and efficient verification processes.

Different sensors including not-very expensive GPS sensors, IMUs, wheel sensor, on-board detection sensors such as cameras and radars are usually fused to obtain an accurate positioning for advanced driver assistance systems and ultimately self-driving vehicles. Meanwhile, the more expensive high-performance positioning systems such as RTK [5] can be used for evaluation and verification purposes. In addition to these high performance sensors, simulation environments and virtual testing are a powerful and necessary tool for testing and verification of components for self-driving vehicles which also includes the positioning components.

A prediction of the sensor error can theoretically be calculated fairly precise with information about all sources together with the satellite positions relative to the receiver. Some of the most critical error sources for a Global Navigation Satellite System (GNSS) are listed in Table 1, where course estimations of the error for each satellite measurement are presented. The *random* column represents errors that could efficiently be decreased by smoothing over approximately 10 seconds. However, a simulation environment for self-driving vehicles seldom has access to information about all sources.

An internal model already exists in the GNSS receiver that estimates the difference between the ideal case where no delays or satellite errors exists and a guess of the real case scenario. These models are vital for the measurement procedure and a popular subject for research since more precise models will increase the accuracy of the GNSS measurements [7], [4]. In [6] characterization and modeling of the ambiguities in internal models has been performed using the pseudo-range equation by which the different sources of errors can be modeled individually. The method proposed in [8] models the complete GNSS functionality. These modeling methods of the receiver ambiguities are comprehensive and can characterize the performance under many conditions. However, the problem with these methods, including [6] and [8] is their equally extensive requirements on the training data, and during simulation, precise knowledge about the conditions. A modeling method suited for the autonomous vehicle industry, that can be used with only lateral and longitudinal positioning errors and optionally different environmental conditions, needs to be developed. This means that only data available in the vehicle should be required.

This paper proposes a modeling method with which the ambiguities of the internal measurement models can be characterized. The main contributions of this paper is the proposed generative model and an efficient learning algorithm to learn the model using large


*SEFAIAS'18, May 28, 2018, Gothenburg, Sweden*

ACM ISBN 978-1-4503-5739-5/18/05…$15.00
https://doi.org/10.1145/3194085.3194089

**Table 1: List of possible error sources and their influence on the pseudo-range in meters. The numbers are coarse estimations and should only be used to give a hint about the importance of different sources. *Random* column represents relatively high error frequencies, while *Bias* represents the low error frequencies [1].**

| Error source | Random | Bias | Total |
|---|---|---|---|
| Ephemeris | 0.1 | 1.0 | 1.0 |
| Clock | 0.4 | 1.0 | 1.2 |
| Ionosphere | 0.4 | 4.0 | 4.0 |
| Troposphere | 0.5 | 0.5 | 0.7 |
| Multipath | 0.3 | 1.0 | 1.4 |
| Receiver noise | 0.2 | 0.5 | 0.5 |
| Raw pseudo-range | 0.8 | 4.4 | 4.6 |

amounts of logged data from Volvo cars. The logged data consists of the coordinates of the vehicles measured using the production sensor, which should be modeled, and a high precision reference sensor that is used as the ground truth. Although GPS error analysis is extensively studied in the literature, this paper is the first, to the best of our knowledge, to propose a model which is suitable in the context of sensor analysis for self-driving vehicles and is structured to capture complex variations over time. We propose an iterative algorithm to learn a model based on autoregression and Gaussian mixture models, where unsupervised data clustering and supervised model fitting are iterated until convergence. Moreover, additional affecting conditions can be included to gain a deterministic control of the characteristics. The model captures the distribution of the absolute error and first difference distribution and is able to generate artificial data with similar properties.

## 2 DATA

The production sensor is an off-the-shelf GPS available at Volvo Cars and data is collected within the Volvo Cars's DriveMe project.

The unit RT3000 from OxTS using RTK GPS is used as the reference sensor which has a specified precision of 0.01 meter CEP (Circular Error Probable) given clear sky. This means that 50 % of all samples should have an error distance less then 1 centimeter in the horizontal plane. The logged RT3000 data comes at 100 Hz while the production GPS, only has an update frequency of 1 Hz.

The reference signal is down-sampled with interpolation to match the timestamps of the production sensor. Occurrences of large errors are manually verified by visual verification. A sequence of training data with an incorrect or non-precise reference is discarded.

In the later description about the model structure, cf Section 3.1, 4 different sub-models are trained on data recorded during different conditions which are listed below.

- Clear sky (highway) and calm ionosphere
- Clear sky (highway) and stormy ionosphere
- Occluded sky (urban) and calm ionosphere
- Occluded sky (urban) and stormy ionosphere

The choice to use these two condition types are based on the error source severity table (see Table 1) and which condition types that are important in a simulation environment.

Data is recorded in the area of Gothenburg only. Ionospheric data is provided by Swepos, Lantmäteriet and whether the data is recorded in a urban environment with extensive occlusion of the sky or not is determined manually afterwards through visual coordinate inspection.

## 3 METHOD

In this section, the proposed method to model the GPS error is explained.

### 3.1 Model Structure

The error is approximated as a stochastic process $F(t)$, where $t$ is time, bounded such that the variance converge when $t \rightarrow \infty$. An approximation of characteristics as one single stationary stochastic process is however not accurate, since the behavior changes over time. By dividing each data set into smaller segments and train separate models on different parts of the data, some of these variations are captured. This section describes the structure of the model for it to be able to capture characteristic variations and for the user to be able to specify current conditions.

The model consists of Top, Middle , and Bottom layers that are described here in details. In the Top layer, which provides an interface to the model, environmental conditions can be specified as input and artificial observations are returned as output. The condition types that are not specified in the input will be selected randomly from the available options. The Middle layer consists of a number of sub-models with different characteristics. The environmental conditions that are available during simulation are used to select a specific sub-model. This means that if an urban environment with high probability for multipaths is specified, a sub-model trained on corresponding data is selected. Only one sub-model is active at a time. Those condition types that are available to control but not specified will be selected randomly over time, based on a transition matrix.

The bottom layer is pure stochastic. Inside each sub-model there exists a number of clusters each containing parameters for describing a stochastic process. During the training phase, data for each sub-model is divided into segments. Segments with similar properties are clustered using Algorithm 1. A complete set of parameters to describe the stochastic behavior for each cluster of data segments are then calculated, see Section 3.2. The segment size is chosen such that each segment should be able to enclose large and slow variations in the error. After analyzing the data visually the size was set to 1000 samples which approximately corresponds to sequences of 17 minutes.

### 3.2 Stochastic Model

The stochastic part of the model, the bottom layer, is modeled as auto-regressive (AR) processes. Lateral and longitudinal error are modeled separately. A general explanatory equation of an AR model can be seen in Eq. 1:

**Data:** Finishing criteria: Number of desired clusters, $N$
**Result:** Cluster labels for each segment, *clusterLbls*
$Nmax \leftarrow$ number of segments in the data;
$clusterLbls \leftarrow 1{:}Nmax$;
/* One cluster for each segments */
$nrOfClusters \leftarrow$ current number of clusters;
**while** $nrOfClusters > N$ **do**
  **for** $i \leftarrow 1$ **to** $nrOfClusters$ **do**
    prop(i) $\leftarrow$ calculate properties for cluster $i$;
    /* Properties: Residual error standard deviation, AR coefficients */
  **end**
  $dists \leftarrow$ calculate Eucliedian distance in property space between clusters;
  $Nmax \leftarrow floor(max(Nmax/2, N))$;
  $clusterLbls \leftarrow$ cluster $clusterLbls$ into $Nmax$ number of clusters;
  /* Using hierarchical clustering with single linkage [3] */
  $nrOfClusters \leftarrow$ number of different clusters in $clusterLbls$.
**end**

**Algorithm 1:** Iterative clustering algorithm

$$y_k = \sum_{i=1}^{p} a_i y_{k-i} + \epsilon_k, \tag{1}$$

where $a_i$ are AR coefficients and $\epsilon_k$ is the residual white noise term. The AR coefficients $a_i$ are estimated using method of moments through Yule-Walker equations [2], which minimizes the residual error in a least square sense.

The properties for the clustering algorithm for the bottom layer was chosen to be the AR coefficients together with the standard deviation of the residual error. This means that the cluster parameters for each complete cluster should be similar to the parameters for each individual segment in the cluster, which also induces similarity of the error characteristics within each cluster.

After completion of the clustering the residual error, $\epsilon$, for each cluster is modeled as a separate Gaussian Mixture Model (GMM) with three components. This choice is based on a comparison between Laplacian and GMMs with different number of components that can be seen in Figure 1.

The influence of the model order, $p$, is evaluated using cross-validation to be able to decide a proper general order for all components in the model. A development set is considered which is not used to optimize the AR parameters, $a_1$, $a_2$, ..., $a_p$. Artificial samples are generated using Eq. 2. With this equation the artificial data is a predicted set of data, but every element is predicted using true historical values. Then, the mean squared error, MSE, (see Eq. 3) of the residual error between the artificial data and the true logged data is computed, where a lower number indicates a better

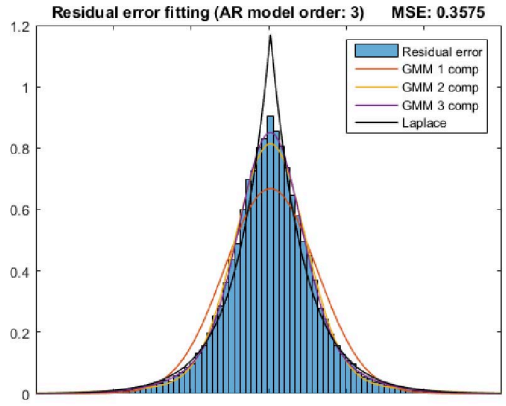


**Figure 1: Four different distribution types are fitted to the residual error distribution that are shown as a histogram.**

**Table 2: Comparison between different orders of the AR model, using Mean Squared Error of generated data compared to the logged data.**

| Order | MSE |
|---|---|
| 1 | 12157.5 |
| 2 | 8903.2 |
| 3 | 8197.6 |
| 4 | 8086.9 |
| 5 | 7908.2 |
| 6 | 7747.2 |
| 7 | 7692.1 |

model.

$$z_k = \begin{cases} y_k, & \text{if } 1 \leq k \leq p \\ \sum_{i=1}^{p} a_i y_{k-i}, & \text{if } p < k \end{cases} \tag{2}$$

where $z_k$ are the artificial samples and $y_k$ are the logged positioning errors.

$$MSE = \frac{1}{N}(\mathbf{y} - \mathbf{z})(\mathbf{y} - \mathbf{z})^T \tag{3}$$

where $\mathbf{y}$ is a $1 \times N$-dimensional logged sequence, $\mathbf{z}$ is the corresponding generated sequence and $T$ denotes the transpose operation.

The selection of model order is a choice between maximizing the performance and minimizing the model complexity as well as the risk for overfitting. The results in Table 2 show that $p = 3$ is a reasonable choice, given the MSE development.

To generate artificial samples from the model, a submodel is chosen based on the input data from the Top layer. For a given submodel, the segment clusters are chosen according to a transition matrix, which is trained from the training data after the cluster parameters are learned. This transition matrix represent how the different segments are clustered within each submodel during training, hence training of such matrix can be performed by simply looping through the segments in a chronological order and count the change of cluster label between segments.

## 4 RESULTS

Figure 2 and 3 compare training and simulated data for the total error distribution and the distribution of the first difference respectively in the lateral direction for one submodel. The error magnitudes are not shown due confidentiality of Volvo Cars data. The distributions of simulated samples correspond very well to the training data. Training data for this submodel is recorded during ideal GNSS conditions when considering ionosphere and surroundings and it contains over 60k samples.

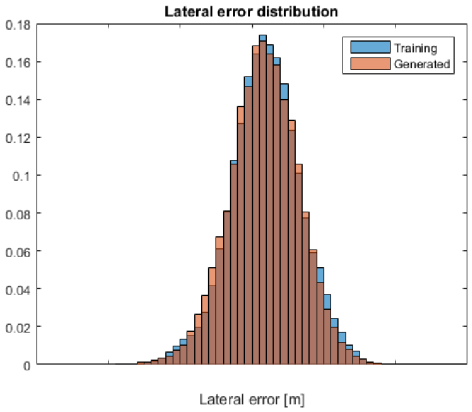


**Figure 2: Total error distribution - Large data set**

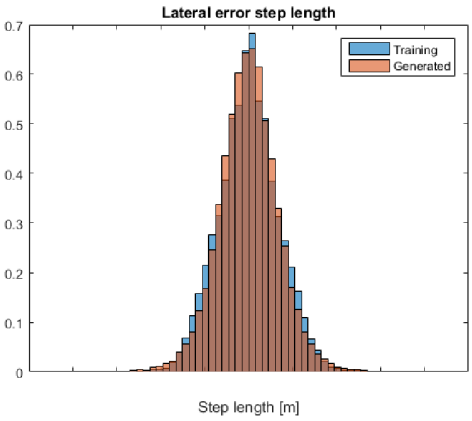


**Figure 3: First difference distribution of the error - Large data set**

Another submodel example with a smaller training data set, approximately 10k samples, recorded in city environment is visualized in Figure 4 and 5. The size of simulated sample set is however 50k. The fitting in Figure 5 is not as good as in Figure 3. Something that may be improved if the amount of training data increases, but in this case it is also possible that a city environment may cause positioning errors with a first difference distribution that are difficult to model with an autoregression and a three component GMM. This could be confirmed by recording more training data.

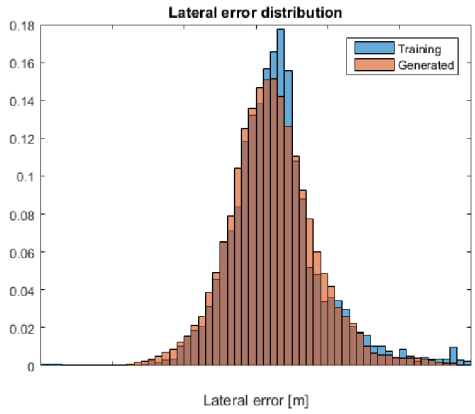


**Figure 4: Total error distribution - Small data set**

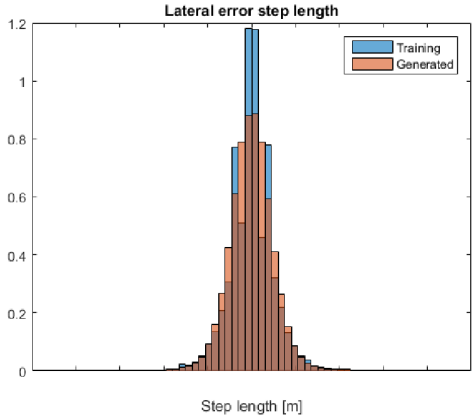


**Figure 5: First difference distribution of the error - Small data set**

Figure 6 shows a simulation sequence, in comparison with a sequence of training data. The data represents the lateral error for submodel number 2 in the resulting model. Note that the training data and simulated error do not have to be correlated within the submodel, only that the two features, total and first difference distributions should be matched.

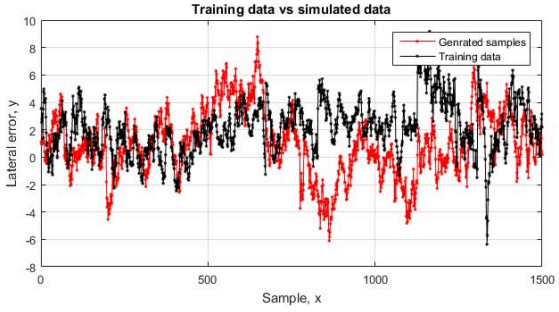


**Figure 6: Lateral error sequence for simulated and training data.**

## 5 ACKNOWLEDGEMENT

This work is supported by the BADA-SEMPA project, which is partially financed by the Swedish government agency Vinnova.

## 6 CONCLUSION

This paper proposes a modeling method for GPS errors that is suitable for the self-driving vehicles development. By providing a large dataset of logged GPS errors and optionally a set of conditions, the error behavior is stochastically modeled. Our proposed solution is a two-step algorithm: in the first step, a submodel is chosen given provided conditions, and in the second step, one of the AR models with the GMM distribution for the residual error within the given submodel is chosen to generate new samples from. Our experiments show that the model successfully generates an output with similar characteristics including the total error distribution and the first difference distribution.

## REFERENCES

[1] Enge and van Diggelen. 2014. Online course: GPS: An Introduction to Satellite Navigation. University Lecture. (2014).
[2] Gidon Eshel. [n. d.]. Course material: The Yule Walker Equations for the AR Coefficients. ([n. d.]).
[3] J. C. Gower and G. J. S. Ross. 1969. Minimum Spanning Trees and Single Linkage Cluster Analysis. *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 18, 1 (1969), 54–64.
[4] Donghyun Kim and Richard B. Langley. 2001. *Estimation of the Stochastic Model for Long- Baseline Kinematic GPS Applications.* Technical Report. University of New Brunswick.
[5] NovAtel. [n. d.]. An introduction to GNSS, Chapter 5 Resolving Errors. ([n. d.]).
[6] James Rankin. 1994. *GPS and Differential GPS: An Error Model for Sensor Simulation.* Technical Report. St. Cloud State University. 260–266 pages.
[7] Jinling Wang et al. 2005. Online Stochastic Modelling for Network-Based GPS Real-Time Kinematic Positioning. *Journal of Global Positioning Systems* 4, 1-2 (2005), 113–119.
[8] Jón Ólafur Winkel. 2003. *Modeling and Simulating GNSS Signal Structures and Receivers.* Ph.D. Dissertation. Bundeswehrs universitet.