2018 IEEE Intelligent Vehicles Symposium (IV)
Changshu, Suzhou, China, June 26-30, 2018

A Statistical GPS Error Model for Autonomous Driving

Erik Karlsson[1] and Nasser Mohammadiha[2]

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 and satellite clock 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. For this purpose, we propose a state-based semi-stochastic model and an efficient learning algorithm, where stochastic parts are modelled using autoregression and Gaussian mixture models. The resulting model successfully mimics the distributions of the absolute error and the first difference, for data with ideal GNSS conditions.**

I. 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 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. However, a simulation environment for self-driving vehicles seldom has access to information about all sources.

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 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.

A. Background and Related Works

The commonly used positioning systems include Global Positioning System (GPS) which is an American version of a Global Navigation Satellite System (GNSS). GLONASS from Russia, Galileo from Europe and BeiDou from China also use very similar systems. In these systems, a pseudo random code is sent from each satellite, while other data such as ephemeris information and time is sent in parallel [1], [2]. With this information an estimated distance to the satellite, the pseudo-range, is calculated. Since the clock in the receiver often is cheap and non-precise the receiver clock error is estimated together with the position in three dimensions. To solve the necessary pseudo range equations and thereby lock the GPS receiver's position, at least four measurements from satellites spread around in the sky are needed. One for each unknown parameter, three for position and one for the clock error.

Some of the most critical error sources for a GNSS are listed in Table I, 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. The Differential GPS (DGPS) was developed to decrease the magnitude of these errors. Several of the sources will affect all nearby receivers similarly such as the ephemeris error, ionosphere, troposphere and satellite clock error. With a stationary GPS receiver with a known position, and thereby a known posi-

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

[1] Erik Karlsson is with ÅF Technology AB, Gothenburg, Sweden. erik.a.karlsson@afconsult.com

[2] Nasser Mohammadiha is with Zenuity AB, Gothenburg, Sweden. nasser.mohammadiha@zenuity.com

978-1-5386-4451-5/18/$31.00 ©2018 IEEE

TABLE I: List of possible error sources and their influence on the pseudo-range in meters. The numbers are course 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 [5].

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

tioning error, a mobile receiver can subtract this error from its own measurements. By doing so the accuracy increases to a few meters [3]. Another system worth mentioning is the Real Time Kinematic (RTK) GPS [4]. This system works as DGPS but uses the carrier wave itself, on which the pseudo random code is encoded, to estimate the distance to the satellite. This method is more complicated and expensive but can improve the precision even further to just a few centimeters [3].

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 [6], [7]. In [8] 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 [9] 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 [8] and [9] 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.

II. 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. Due confidentiality of Volvo cars data most plots have their scales removed.

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. It is also equipped with accelerometers and gyros to decrease the noise level. 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 III-A, 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 I) and which condition types that are important/accessible in a simulation environment.

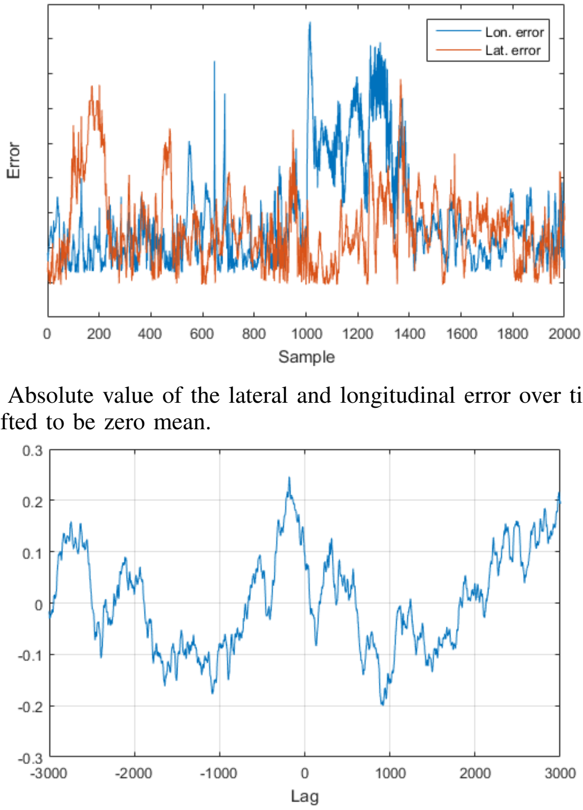
Data is recorded in the area of Gothenburg only. Ionospheric data is provided by Swepos, Lantmäteriet and whether the data is recorded in a urban environment with extensive occlusion of the sky or not is determined manually afterwards through visual coordinate inspection. Data is recorded during several trips, often not more than 1 hour each. In total approximately 40 hours of collected data is used.

The correlation between lateral and longitudinal error is analyzed to evaluate the importance of including such correlation in the model. The correlation can be viewed in Figure 1. The analysis is performed on the absolute value of the error, shifted to be zero mean. A low frequency correlation will create a wide peak at zero lag in the analysis and a higher frequency correlation will induce a more narrow peak. Figure 1 contains a wide peak which indicates low frequency correlation and the highest value is located with a lag of 200, which in this case corresponds to 200 seconds. A correlation with 200 seconds shift between lateral and longitudinal error is unreasonable and it is therefore possible to conclude that the high frequency correlation is relatively low, and that the small variations superimposed on the wide peak are consequences from noise. The lower frequency correlation can be explained as consequences of changes in the surroundings. For example changes caused by the receiver moving into an area with high buildings. In such cases both lateral and longitudinal error may increase for longer periods.

After analysis the high frequency correlation could intentionally be excluded from the model. Low frequency correlation will be included by using the method presented in Section III-A.

III. Method

In this section, the proposed method to model the GPS error is explained. The model structure is presented in Section III-A, which is followed by a description of the stochastic model in Section III-B and then more practical aspects are discussed in Section III-C and III-D.



(a) Absolute value of the lateral and longitudinal error over time, shifted to be zero mean.

(b) Lateral and longitudinal correlation.

Fig. 1: Lateral and longitudinal analysis.

A. 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 three layers described below:

1) Top layer: Model interface. Environmental conditions can be specified as input and artificial observations are generated as output.

2) Middle layer: 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.

3) Bottom layer: 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 III-B.

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.

```
**Input:** Number of desired clusters, $N$
**Output:** 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 [10]                     */
    $nrOfClusters \leftarrow$ number of different clusters in
      $clusterLbls$.
**end**
```

Algorithm 1: Iterative clustering algorithm

B. 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:

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

where a_i are AR coefficients and ϵ_k is the residual white noise term. The AR coefficients a_i are estimated using method of moments through Yule-Walker equations [11], 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.

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

Order	MSE
1	0.4267
2	0.3177
3	0.3038
4	0.3009
5	0.2923
6	0.2895

After completion of the clustering the residual error, ϵ, 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 2.

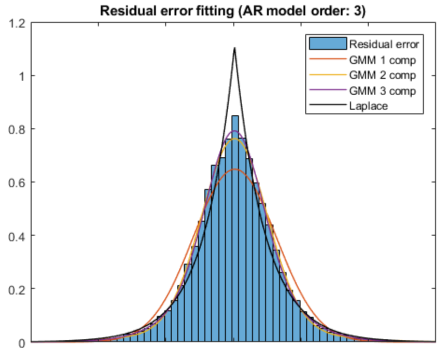


Fig. 2: Four different distribution types are fitted to the residual error distribution that are shown as a histogram.

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 model.

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

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 trade-off between maximizing the performance and minimizing the model complexity as well as the risk for overfitting. The results in Table II show that $p = 3$ is a reasonable choice. Higher orders decrease the MSE with relative small differences. The MSE values are scaled due confidentiality of Volvo cars data.

C. Solving Practical Training Data Problems

As described in Section II non-precise training data has to be discarded. Such data treatment as well as merging data sets to increase the amount of training data will induce transitions in the training data, which are not members of the real first difference distribution. To solve this the data is split at each transition as well as at evenly distributed points based on the specified segment size. Consecutive parts within the specified segment are then pre-clustered before the recursive algorithm begins. With this method, all unwanted transition can be neglected while the order within decomposed segments remains.

D. Simulation and Model Functionality

The condition types that are not specified from the simulation environment will be selected randomly through a set of transition matrices, one for each condition, reflecting the behavior in the logged data. Then, for a given submodel the segment cluster is chosen randomly after a transition matrix trained automatically. This transition matrix represent how the different segments are clustered within each submodel during training. The update period when using the cluster transition matrix is set by the mean length of the segments. Lateral and longitudinal data is divided identically into segment and clusters. By always using the same segment cluster for both lateral and longitudinal data during simulation, low frequency correlation between the two signals in the training data can be regenerated in the simulated data.

IV. Results

Evaluation of time-series models is a challenging task and choosing proper evaluation metrics/criteria strongly depends on the application. Here, to evaluate the proposed error model, we compare the total error distribution and the distribution of the first difference between the real data and artificially generated data from the model. Using these two criteria helps us to evaluate the capacity of the model to generate errors with similar magnitudes and error trajectories with similar temporal characteristics as the real data.

All available data, i.e., the whole training dataset that was used to optimizes the model, has been used here in order to include as many of the rare errors as possible in the evaluation. By doing so, we are measuring the expressiveness of the model and model capacity to fit a given dataset, which is of interest for our application, rather than the prediction capacity of the model which is typically done, e.g., in classification tasks.

Figure 3 and 5 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. In many applications the most important property to capture is the tails in Figure 3, since these describe the occurrence of large magnitude errors, the anomalies. Figure 4 contains the same data as Figure 3 but is zoomed in for clearer visualization of small values (note the different scaling compared to Figure 3). Training data for this submodel is recorded during ideal GNSS conditions when considering ionosphere and surroundings and it contains over 60k samples.

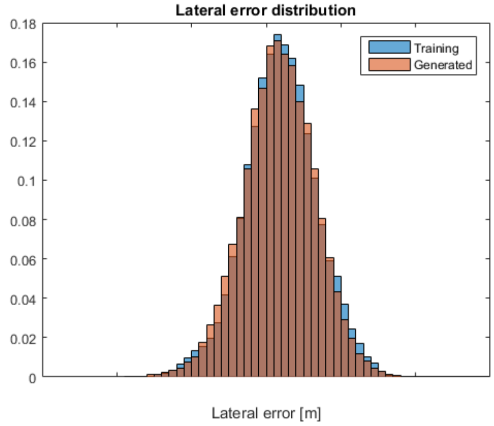


Fig. 3: Total error distribution - Large data set

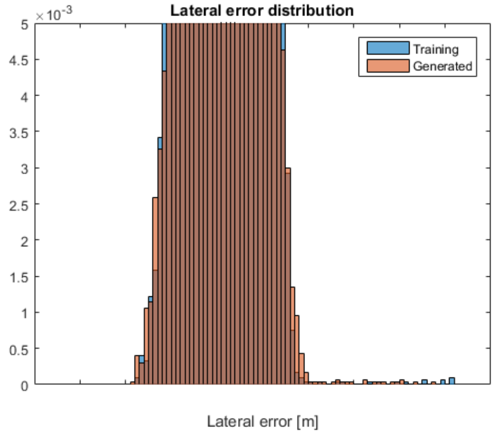


Fig. 4: Total error distribution (different axis scaling) - Large data set

Another submodel example with a smaller training data set, approximately 10k samples, recorded in city environment is visualized in Figure 6 and 7. The size of simulated sample set is however 50k. The fitting in Figure 7 is not as good as in Figure 5. 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 8 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.

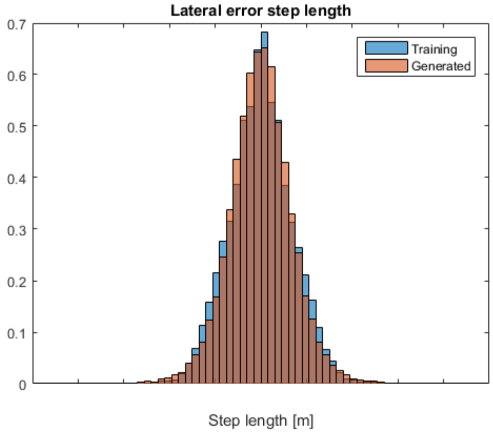


Fig. 5: First difference distribution of the error - Large data set

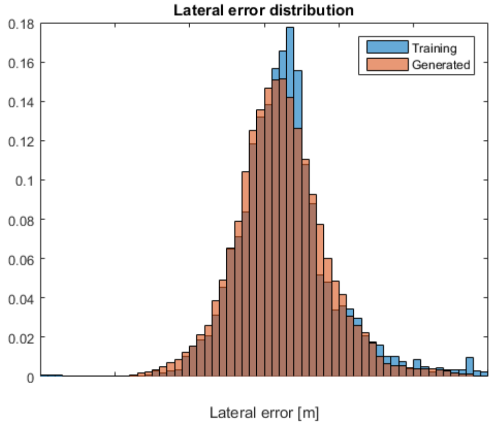


Fig. 6: Total error distribution - Small data set

As mentioned earlier, when simulating without specifying a desired submodel, the simulation will switch between all available submodels according to the specified submodel probabilities. Results of a simulation can be seen in Figure 9.

As a final point, we look into the model convergence and required training data, which intuitively is strongly dependent on the number of submodels and how these are divided into segments. Figure 10 and 11 show the convergence progress for a large cluster of segments with approximately 50k samples.

In our specific case with a receiver giving measurements at 1 Hz, it seems to be preferable to have at least 10k to 15k samples in each cluster of training data.

V. 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

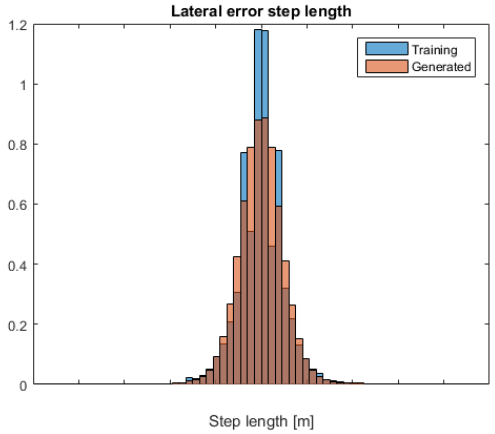


Fig. 7: First difference distribution of the error - Small data set

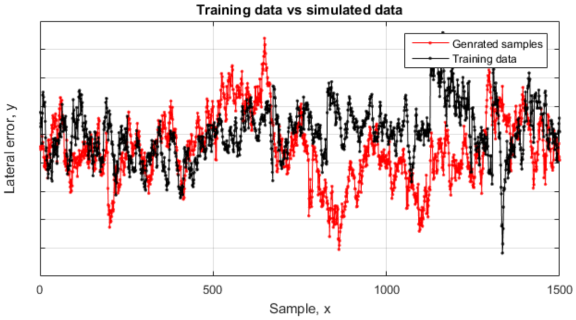


Fig. 8: Lateral error sequence for simulated and training data.

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] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins, *Global Positioning System - Theory and Practice*. Wien: Springer, 1997.

[2] J. B.-Y. Tsui, *Fundamentals of Global Positioning System Receivers - A Software Approach*. New York: Wiley Interscience, 2000.

[3] GMV, "GPS performances - navipedia," 2011. [Online]. Available: http://www.navipedia.net/index.php/GPS_Performances

[4] NovAtel, "An introduction to GNSS, chapter 5 resolving errors." [Online]. Available: https://www.novatel.com/an-introduction-to-gnss/

[5] Enge and van Diggelen, "Online course: GPS: An introduction to satellite navigation," University Lecture, 2014.

[6] J. Wang *et al.*, "Online stochastic modelling for network-based GPS real-time kinematic positioning," *Journal of Global Positioning Systems*, vol. 4, no. 1-2, pp. 113–119, 2005.

[7] D. Kim and R. B. Langley, "Estimation of the stochastic model for long- baseline kinematic GPS applications," University of New Brunswick, Tech. Rep., 2001.

[8] J. Rankin, "GPS and differential GPS: An error model for sensor simulation," St. Cloud State University, Tech. Rep., 1994.

[9] J. O. Winkel, "Modeling and simulating gnss signal structures and receivers," Ph.D. dissertation, Bundeswehrs universitet, 2003.

[10] Cambridge University Press, "17 hierarchical clustering," 2009. [Online]. Available: https://nlp.stanford.edu/IR-book/pdf/17hier.pdf

[11] G. Eshel, "Course material: The Yule Walker equations for the AR coefficients."

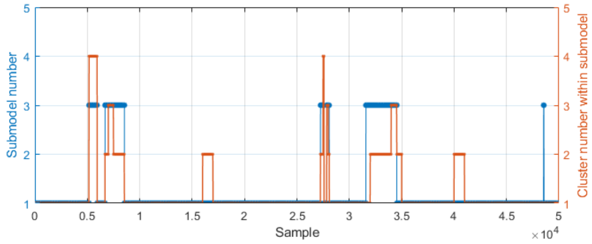


Fig. 9: Submodel and cluster number within submodel over time. Note that the sample time is 1 second.

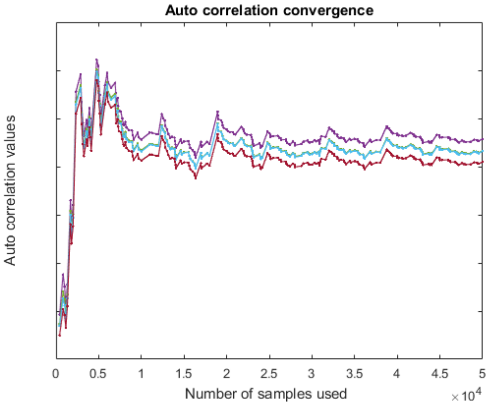


Fig. 10: Convergence of the 4 auto-correlation coefficients used for a 3rd order autoregression model.

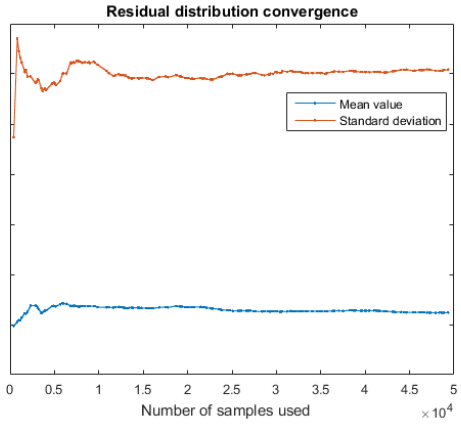


Fig. 11: Convergence of the residual error expressed with a fitted Gaussian distribution.