Systems and Computers in Japan, Vol. 37, No. 1, 2006
Translated from Denshi Joho Tsushin Gakkai Ronbunshi, Vol. J87-D-II, No. 6, June 2004, pp. 1199–1207

Adjustment of Sampling Locations in Rail-Geometry Datasets: Using Dynamic Programming and Nonlinear Filtering

Masako Kamiyama[1,2] and Tomoyuki Higuchi[3]

[1]Track Technology Division, Railway Technical Research Institute, Kokubunji, 185-8540 Japan

[2]Department of Statistical Science, The Graduate University for Advanced Studies (Sokendai), Tokyo, 105-8569 Japan

[3]Department of Statistical Modeling, The Institute of Statistical Mathematics, Research Organization of Information and Systems, Tokyo, 106-8569 Japan

SUMMARY

A track inspection car, which measures the shape of railway tracks (hereafter, rail geometry) while it is running on rails, discretizes the measurement results at nearly fixed spatial intervals. However, the distance between the discretized locations (spatial sampling intervals) may shorten or lengthen locally due to slipping or sliding of the car wheel, and this prevents the sampling locations from aligning with those of a dataset obtained with another measuring run. The authors developed an algorithm for approximately aligning the sampling locations of the measurement datasets obtained with different runs. First, they considered this problem as the selection of the series of data corresponding to each supervised data from a training dataset, which was constructed by interpolation in order to minimize the evaluation function of a number sequence representing data points. Next, they used the maximum likelihood method to identify the unknown parameters contained in the evaluation function. This problem uses two features of the evaluation function. The first is that the evaluation function is minimized by dynamic programming, and the obtained optimum sequence is equivalent to a maximum a posteriori (MAP) estimate in the Bayesian framework. The second is that by converting the evaluation function to a general state space representation, the log likelihood of the model that includes the parameters is obtained by a nonlinear filtering method. Also, to simplify the search for the identification, they devised a parameter search procedure for the parameters in the autoregressive (AR) model. © 2005 Wiley Periodicals, Inc. Syst Comp Jpn, 37(1): 61–70, 2006; Published online in Wiley InterScience (www.interscience.wiley.com). DOI 10.1002/scj.20313

Key words: dynamic programming; state space representation; nonlinear filter; maximum likelihood method.

1. Introduction

The special train called a track inspection car measures the shape of railway tracks (rail geometry) regularly since this geometry varies slightly under the load of daily passing trains. The measurement results are discretized at equidistant intervals and stored for processing by computers. To obtain temporal variations of the rail geometry, the measurement results ideally should always be discretized at the same locations on the track. However, since the pulse signals for specifying the sampling locations are linked with the rotation of the wheel of the track inspection car as shown in Fig. 1, the pulse generation locations cannot be reproduced. Therefore, the sampling locations on the track

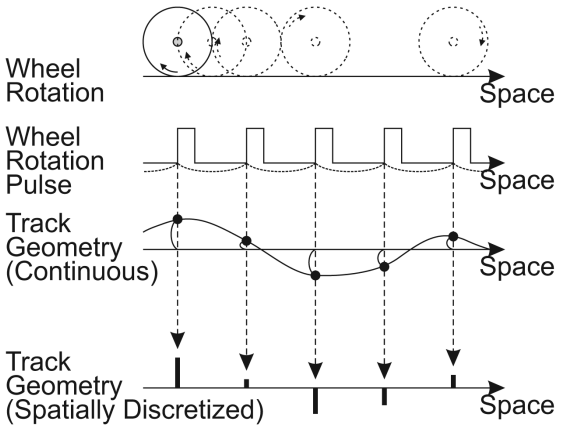

Fig. 1. Scheme for spatial discretization of measured railway track geometry according to wheel rotation pulses.

vary for each measuring run. To align two sets of sampling locations obtained with difference runs, we must estimate the location gaps from the discretized datasets, but this is difficult. If the spatial sampling intervals between all of the data were uniform, these gaps between two sets of the sampling locations could be easily estimated by calculating the cross correlation coefficient. However, because the sampling intervals locally shorten or lengthen as shown in Fig. 2 due to the slipping or sliding of the car wheel, this method cannot be applied. (In Fig. 2, the measurement results are discretized once per rotation of the wheel to simplify the explanation.)

Figure 3 shows an example of the measurement results. The datasets (A) and (B) for the upper two graphs show the values of the track gauge (spacing between the left and right rails) for the same railway section measured 1 day

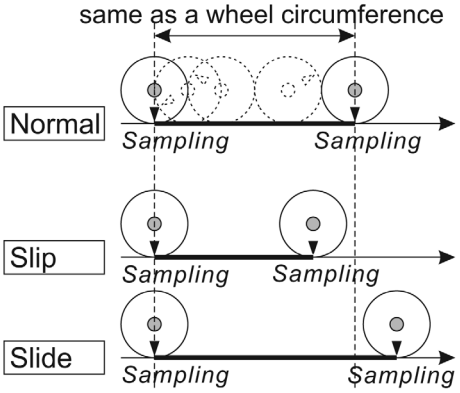

Fig. 2. Scheme for variations in spatial sampling intervals accompanying wheel rotation states.

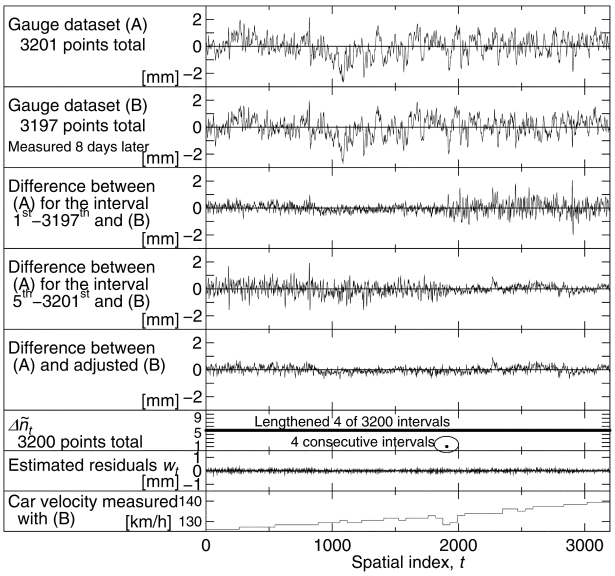

Fig. 3. Example of datasets obtained from two measurements for the same railway section and results of adjusting sampling locations.

and 8 days later (0 means that the gauge is equal to the reference value). Both datasets were discretized at equal intervals (approximately 30 cm; 8 measurements per wheel rotation). In addition, for both (A) and (B), the location where the first data was obtained is somewhere within one sampling interval in the car movement direction of a certain device that is fixed at a specific location on the track. Similarly, the data for the 3201st point of (A) and 3197th point of (B) were also measured within one sampling interval in the car movement direction from another location detection device.

It is known empirically that the gauge hardly varies over an 8-day period or so, and the waveforms for (A) and (B), which are shown in the first and second graphs from the top of Fig. 3, are actually similar. Also, the datasets close to the 1st data in (A) and (B) are obtained at similar locations. This is apparent from the difference between the two waveforms, which is shown in the third graph from the top of Fig. 3. Similarly, the fourth graph from the top of Fig. 3 shows that the datasets close to the end data in (A) and (B) are also obtained at similar locations. In other words, despite the fact that (A) and (B) are measurement values of the gauge for the same railway section, the number of data points constituting (B) is smaller than the number constituting (A) by 4 points (corresponding to approximately 1 m). This is because the sampling interval shortened or lengthened locally due to slipping or sliding of the car wheel during one or the other measuring runs.

We developed an algorithm that uses dynamic programming (DP [1]) and nonlinear filtering for a general state space representation to approximately align the sampling locations for datasets of this kind. The fifth graph in Fig. 3 shows the results when the sampling locations for dataset (B) were aligned with those of dataset (A) by using the proposed algorithm. This paper describes the location adjustment algorithm in Sections 2 and 3, presents a discussion in Section 4, and presents conclusions in Section 5.

2. Adjustment of Sampling Locations by Using Dynamic Programming and Nonlinear Filtering

An overview of the adjustment algorithm that we developed is presented below:

(1) Model a mechanism to be empirically considered to yield the nonuniform sampling.

(2) Formulate this model in a nonlinear optimization problem that can be solved by dynamic programming. However, since this form contains unknown parameters, the solution cannot be uniquely obtained.

(3) Represent the above nonlinear form with a general state space representation and identify the included unknown parameters through a nonlinear filtering algorithm based on the maximum likelihood method.

(4) Substitute the estimated parameters to solve the optimization problem of step (2).

(End)

These steps are explained in detail below.

2.1. Modeling the problem

We decided to consider this location adjustment problem as a problem for selecting the most suitable data points corresponding to individual supervised data $\{Y_1, Y_2, \ldots, Y_{T-1}, Y_T\}$ (where T is the number of the supervised data) from the training dataset that was interpolated so that the number of points was a multiple of α as shown in Fig. 4. Since the gauge dataset is smoothed by an analog low-pass filter before the spatial discretization, the original training dataset is discretized so that the original spatial frequency component does not change. The spatial frequency characteristic of this filter varies with time, since this characteristic changes with the speed of the inspection car. The interpolated dataset hereafter is called the "training dataset $\{X_1, X_2, \ldots, X_{N-1}, X_N\}$." Note that $N (\approx \alpha T)$ is the number of training data after interpolation.

To distinguish between the dataset as a set and the individual data, we hereafter denote $\{Y_1, Y_2, \ldots, Y_{T-1}, Y_T\}$ collectively as $\mathbf{Y}_{1:T}$. The other variables are denoted in a similar manner.

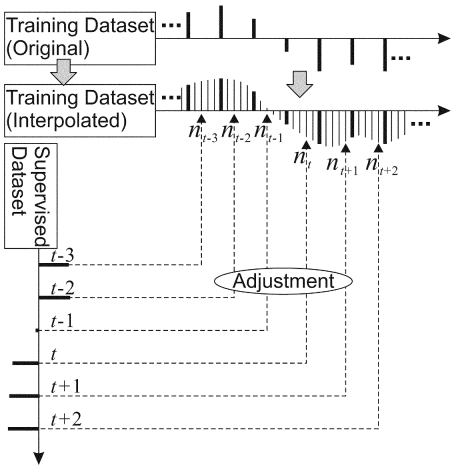

Fig. 4. Scheme for the basic idea for adjusting the discretized location gaps.

Let n'_t be a data point index of the training set selected to match the t-th supervised data Y_t; the value of the corresponding data is $X_{n'_t}$. This problem becomes a problem of finding the optimal number sequence (this is denoted as $\mathbf{n}_{1:T}$) from among the feasible number sequences $\mathbf{n}'_{1:T}$. Also, let $Y_t - X_{n_t}$ be denoted by e_t ($e_t \sim N(0, \tau^2)$).

Next, we discuss the characteristics of $\mathbf{n}_{1:T}$. Although the accurate distribution of the sampling interval is unknown, the running distance due to slipping or sliding is empirically determined to be approximately 0.1% of the total distance or less. Therefore, we assume that $\Delta n_t \stackrel{\text{def}}{=} n_t - n_{t-1}$ of $\mathbf{n}_{1:T}$, which corresponds to the sampling interval, is α for almost every t. Also, for the range of values of Δn_t, this article employs $1 \le \Delta n_t \le 2\alpha - 1$. This is based on the empirical knowledge that was obtained from conventional running experiments. Add to this assumption, we decide that the probability distribution of Δn_t would be constant for all values other than α.

We also assume that the second-order difference $\Delta^2 n_t \stackrel{\text{def}}{=} n_t - 2n_{t-1} + n_{t-2}$ of $\mathbf{n}_{1:T}$ is 0 for almost every t. This is because $\Delta^2 n_t = 0$ is naturally zero when the wheel rotates regularly; $\Delta n_t = \Delta n_{t-1} = \alpha$. Similarly, even when the wheel slips or slides, the wheel in the course of slipping or sliding seems to continue rotating at the same rate ($\Delta^2 n_t = 0$), or to recover normal rotation ($\Delta n_t = \alpha$).

2.2. Adjusting the location by using dynamic programming

From the discussion above, we model the characteristics of the number sequence $\mathbf{n}_{1:T}$ indicating the optimal sequence of the data points as follows:

- $\Delta n_t \stackrel{\text{def}}{=} n_t - n_{t-1}$ is α for almost every t.
- $\Delta^2 n_t \stackrel{\text{def}}{=} n_t - 2n_{t-1} + n_{t-2}$ is 0 for almost every t.

- $e_t \sim N(0, \tau^2)$ (where $e_t = Y_t - X_{n_t}$).

At this time, $\mathbf{n}_{1:T}$ is obtained by optimizing the following evaluation function $\sum F(n_t')$:

$$\begin{aligned}\mathbf{n}_{1:T} &= \arg\min \sum_{t=1}^{T} F(n_t') \\ &= \arg\min_{\mathbf{n}'_{1:T}} \left\{ \sum_{t=1}^{T} (Y_t - X_{n_t'})^2 \right. \\ &\quad + \mu_1 \sum_{t=2}^{T} \xi(\Delta n_t' - \alpha, \alpha - 1) \\ &\quad \left. + \mu_2 \sum_{t=3}^{T} \xi(\Delta^2 n_t', 2\alpha - 2) \right\} \end{aligned} \quad (1)$$

where

$$\xi(n, \gamma) = \begin{cases} 0 & (n = 0) \\ 1 & (0 < |n| \le \gamma) \\ \infty & (\gamma < |n|) \end{cases} \quad (2)$$

$$\mu_1 \ge 0, \quad \mu_2 \ge 0 \quad (3)$$

$$n_t' \in S_t \quad (S_t \text{ is the search area given for } n_t') \quad (4)$$

The factors $\mu_1 \ge 0$ and $\mu_2 \ge 0$ are penalties that are added to the evaluation function when $\Delta n_t' \ne \alpha$ and $\Delta^2 n_t' \ne 0$ respectively.

In addition, as mentioned above, the data points at both ends of the supervised and training datasets are always measured within a fixed distance from specific locations on the track. Therefore, the boundary conditions for $\mathbf{n}_{1:T}$ are as follows:

$$n_1 \in S_1 = \{1, 2, \cdots, 2\alpha - 2, 2\alpha - 1\}$$

$$n_T \in S_T = \{N - (2\alpha - 1) + 1, \cdots, N - 1, N\}$$

Assume that both the penalties n_1 and n_T have uniform distributions.

For obtaining the $\mathbf{n}_{1:T}$ that satisfies the conditions shown in (1) to (4), "dynamic programming" (DP) can be applied [1], since this optimization problem observes the principle of optimality. Substitute $\mathbf{x}_t' \stackrel{\text{def}}{=} [n_t' n_{t-1}']^T$ (where T represents transpose) in Eq. (1) and let

$$\begin{aligned} f(\mathbf{x}_t', \mathbf{x}_{t-1}') \stackrel{\text{def}}{=} & (Y_t - X_{n_t'})^2 + \mu_1 \xi(\Delta n_t' - \alpha, \alpha - 1) \\ & + \mu_2 \xi(\Delta^2 n_t', 2\alpha - 2) \end{aligned}$$

then $g(\mathbf{x}_t') \stackrel{\text{def}}{=} \sum_{j=1}^{t} F(n_j')$ can be replaced by the following recursive formulation:

(1) Calculate $g(\mathbf{x}_t') = \min[g(\mathbf{x}_{t-1}') + f(\mathbf{x}_t', \mathbf{x}_{t-1}') | \mathbf{x}_{t-1}' \in \mathbf{S}_{t-1}]$ for each element $\mathbf{x}_t' \in \mathbf{S}_t$ (where $\min[\cdot]$ represents the minimum value of the elements within $[\cdot]$).

(2) Increment t by 1 and return to step (1).

where $\mathbf{S}_t \stackrel{\text{def}}{=} [S_t S_{t-1}]^T$. When $g(\mathbf{x}_t')$ is regarded as a state that transitions as t increases, the value of $g(\mathbf{x}_t')$ depends only on $g(\mathbf{x}_{t-1}')$, $\mathbf{x}_t'$, and $\mathbf{x}_{t-1}'$ alone.

(End)

Since $\mathbf{n}_{1:T}$ varies with the values of parameters μ_1 and μ_2, we should carefully determine the parameter values.

2.3. Dynamic programming and MAP estimation in the Bayesian framework

To determine the values of μ_1 and μ_2, $\mathbf{n}_{1:T}$, which was obtained in the previous section, should be interpreted statistically. Assume that $e_t = Y_t - X_{n_t}$ is a normal random variable with a zero mean and a standard deviation τ. By multiplying (1) by $-1/(2\tau^2)$ and exponentiating it, we obtain

$$\begin{aligned}\mathbf{n}_{1:T} = \arg\max_{\mathbf{n}'_{1:T}} & \left\{ \prod_{t=1}^{T} \exp\left[-\frac{1}{2\tau^2} e_t^2 \right] \right\} \\ & \cdot \exp\left\{ -\frac{1}{2\tau^2} \left[\mu_1 \sum_{t=2}^{T} \xi(\Delta n_t' - \alpha, \alpha - 1) \right.\right. \\ & \left.\left. + \mu_2 \sum_{t=3}^{T} \xi(\Delta^2 n_t', 2\alpha - 2) \right] \right\} \end{aligned} \quad (5)$$

The former term of the right side (first { }) can be interpreted as a certain multiple of the conditional density $p(\mathbf{Y}_{1:T} | \mathbf{n}'_{1:T})$ of the supervised dataset $\mathbf{Y}_{1:T}$ assuming normality when all $\mathbf{n}'_{1:T}$ are given. Similarly, the latter term (second { }) can also be interpreted as a certain multiple of the prior distribution $p(\mathbf{n}'_{1:T})$ of $\mathbf{n}'_{1:T}$ in the Bayesian framework. Therefore, Eq. (1) can be interpreted as a search for the $\mathbf{n}_{1:T}$ that maximizes the posterior distribution $p(\mathbf{n}'_{1:T} | \mathbf{Y}_{1:T}) \propto p(\mathbf{Y}_{1:T} | \mathbf{n}'_{1:T}) p(\mathbf{n}'_{1:T})$ [this is called the maximum a posteriori (MAP) estimate]. Therefore, μ_1, μ_2, and τ^2 are hyperparameters (in the Bayesian framework) that give the prior distribution [2]. Optimal solutions that are obtained by dynamic programming, not just for this problem, can be interpreted as MAP estimates [3].

Hence, the hyperparameter values can be evaluated using the log likelihood $LL(\mu_1, \mu_2, \tau^2) \stackrel{\text{def}}{=} \log p(\mathbf{Y}_{1:T} | \mu_1, \mu_2, \tau^2)$ of $\mathbf{Y}_{1:T}$. In other words, the hyperparameters for maximizing LL (denoted by $\tilde{\mu}_1$, $\tilde{\mu}_2$, and $\tilde{\tau}^2$) are interpreted as the optimal hyperparameters within the maximum likelihood method. The method for computing LL is described in the next section.

2.4. Estimating hyperparameters by the maximum likelihood method

Log likelihood $LL(\mu_1, \mu_2, \tau^2)$ is obtained as the following sum of conditional probability distributions:

$$LL(\mu_1, \mu_2, \tau^2) = \sum_{t=1}^{T} \log p(Y_t|\mathbf{Y}_{1:t-1}, \mu_1, \mu_2, \tau^2)$$

Therefore, to obtain the *LL*, we should obtain $p(Y_t|\mathbf{Y}_{1:t-1}, \mu_1, \mu_2, \tau^2)$ for all $t \in \{1, 2, \ldots, T\}$.

Next, to obtain $p(Y_t|\mathbf{Y}_{1:t-1}, \mu_1, \mu_2, \tau^2)$, we transform the statistical model (5) into a generalized state space representation [4]. By letting $\mathbf{x}_t \stackrel{\text{def}}{=} [n_t n_{t-1}]$ and $y_t \stackrel{\text{def}}{=} Y_t$, we can obtain the statistical model as follows.

[Observation model]

$$y_t = h(\mathbf{x}_t) + e_t \stackrel{\text{def}}{=} X_{n_t} + e_t$$

where $e_t \sim N(0, \tau^2)$.

[System model]

$$\mathbf{x}_t = f(\mathbf{x}_{t-1}, v_t(\mathbf{x}_{t-1})) \stackrel{\text{def}}{=} \begin{bmatrix} n_{t-1} + \alpha + v_t(n_{t-1}, n_{t-2}) \\ n_{t-1} \end{bmatrix}$$

where $v_t \sim q(\cdot|\mathbf{x}_{t-1}, \mu_1, \mu_2, \tau^2)$ and the distribution of $q(\cdot|\mathbf{x}_{t-1}, \mu_1, \mu_2, \tau^2)$ is as follows.

- When $n_{t-1} - n_{t-2} = \alpha$, then

$$q(v_t|\mathbf{x}_{t-1}, \mu_1, \mu_2, \tau^2) \stackrel{\text{def}}{=} \begin{cases} \frac{1}{\beta_1} & (\text{when } v_t = 0) \\ \frac{\exp\left(-\frac{\mu_1+\mu_2}{2\tau^2}\right)}{\beta_1} & (\text{when } 1 \le |v_t| \le \alpha - 1) \end{cases}$$

where

$$\beta_1 = 1 + 2(\alpha - 1)\exp\left(-\frac{\mu_1 + \mu_2}{2\tau^2}\right)$$

since $\sum q(v_t) = 1$.

- When $n_{t-1} - n_{t-2} \neq \alpha$, then

$$q(v_t|\mathbf{x}_{t-1}, \mu_1, \mu_2, \tau^2) \stackrel{\text{def}}{=} \begin{cases} \frac{\exp\left(-\frac{\mu_2}{2\tau^2}\right)}{\beta_2} & (\text{when } v_t = 0) \\ \frac{\exp\left(-\frac{\mu_1}{2\tau^2}\right)}{\beta_2} & (\text{when } v_t = n_{t-1} - n_{t-2} - \alpha) \\ \frac{\exp\left(-\frac{\mu_1+\mu_2}{2\tau^2}\right)}{\beta_2} & (\text{when } 1 \le |v_t| \le \alpha - 1 \text{ and } v_t \text{ satisfies } v_t \neq n_{t-1} - n_{t-2} - \alpha) \end{cases}$$

where

$$\beta_2 = \exp\left(-\frac{\mu_1}{2\tau^2}\right) + \exp\left(-\frac{\mu_2}{2\tau^2}\right) + [2(\alpha - 1) - 1]\exp\left(-\frac{\mu_1 + \mu_2}{2\tau^2}\right)$$

since $\sum q(v_t) = 1$. Figure 5 shows sample distributions for $q(\cdot)$.

If we assume that e_t and v_t are white noises, we can define the prediction and filtering operations as follows. The $p(y_t|\mathbf{y}_{1:t-1})$ values (where $t = 1, 2, \ldots, T-1, T$) are obtained as the by-product of the computation for $p(\mathbf{n}_{1:T}|\mathbf{y}_{1:T})$, and the log likelihood *LL* is obtained as the sum of their logarithms [4]. Since $\partial Y_t/\partial y_t = 1$, $p(y_t|\mathbf{y}_{1:t-1})$ is equal to $p(Y_t|\mathbf{Y}_{1:t-1})$.

[Prediction]

$$p(\mathbf{x}_t|\mathbf{y}_{1:t-1}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1}) p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}) d\mathbf{x}_{t-1}$$

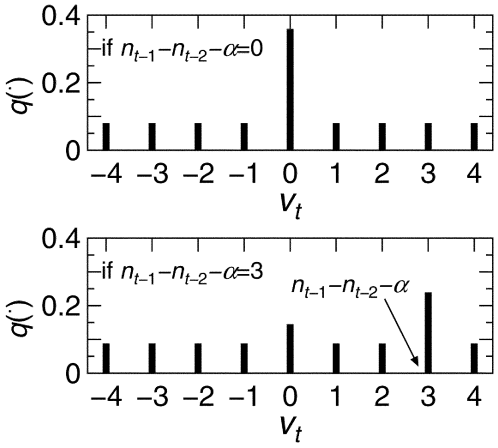

Fig. 5. Typical distribution $q(\cdot)$ for system noise v_t when $\alpha = 5$, $\mu_1 = 1$, $\mu_2 = 2$, and $\tau^2 = 1$.

[Filtering]

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) = \frac{p(y_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1})}{\int p(y_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1})d\mathbf{x}_t} = \frac{p(y_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1})}{p(y_t|\mathbf{y}_{1:t-1})}$$

$p(\mathbf{x}_1|\mathbf{y}_{0:0})$ is a uniform distribution from 1 to $(2\alpha - 1)$.

Some filtering operations in a generalized state space representation involve complicated numerical integration. However, since the system vector $\mathbf{x}_t$ consists of discrete values, while y_t are continuous real values in this model, $p(y_t|\mathbf{y}_{1:t-1})$ can be computed arithmetically without integration. Note that we used a grid search to maximize $LL(\mu_1, \mu_2, \tau^2)$.

We now substitute ($\tilde{\mu}_1$, $\tilde{\mu}_2$, and $\tilde{\tau}^2$) in Eq. (1) and denote the $\mathbf{n}_{1:T}$ obtained by using dynamic programming to optimize that equation by $\tilde{\mathbf{n}}_{1:T}$ hereafter.

2.5. Results and discussion

Table 1 shows the parameters that were estimated by the algorithm described above when the actual measurement values for (A) shown in Fig. 3 were taken as the supervised dataset and the values for (B) were taken as the training dataset (before adjustment). (Note that $\alpha = 5$ is used in this paper.) Also, $\tilde{\mathbf{n}}_{1:T}$ that was estimated by dynamic programming is shown as $(\mathbf{Y}_{1:T} - \mathbf{X}_{\tilde{\mathbf{n}}_{1:T}})$ in the fifth graph and as $\Delta\tilde{n}_t$ in the sixth graph from the top of Fig. 3. From the sixth graph, it is apparent that if it is assumed that the track inspection car did not slip or slide during the measurement run for dataset (A), then it is assumed to have slid for a distance corresponding to four points (approximately 1 m) near $t = 1900$ during the measurement run for dataset (B). Also, the track inspection car's running speed when measuring dataset (B), which is shown in the bottom graph in Fig. 3 (this is calculated back from the number of distance pulses generated within a unit time and normally cannot be referred), reveals that the train's speed dropped unnaturally in the vicinity of where the train was assumed to have slid and suggests that it is highly likely that the wheel actually slid and the generation of distance pulses was temporarily reduced.

Next, Fig. 6 shows the results when $p(\mathbf{x}_t|\mathbf{y}_{1:T})$ is obtained by fixed-interval smoothing as follows:

Table 1. Estimated parameters obtained using the actual measurement values in Fig. 3 (characterizing the differences $\tilde{\mathbf{e}}_{1:T}$ as a Gaussian white noise sequence)

$\tilde{\mu}_1$	$\tilde{\mu}_2$	$\tilde{\tau}^2$	LL	$\Delta\tilde{n}_t \neq \alpha$
0.42	0.43	0.0310	608	4

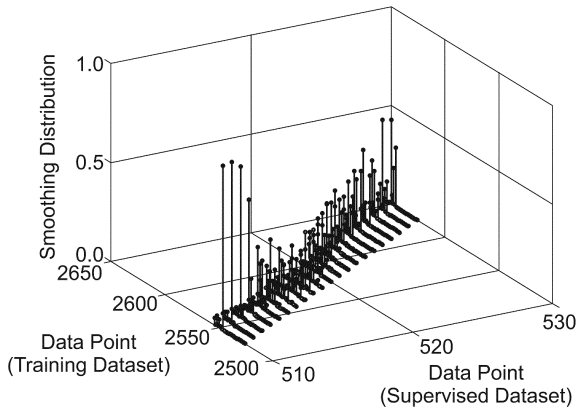

Fig. 6. Fixed-interval smoothing distributions $p(n_t|\mathbf{y}_{1:T})$ (closeup view of $t \approx 520$) (characterizing the differences $\tilde{\mathbf{e}}_{1:T}$ as a Gaussian white noise sequence).

$$p(\mathbf{x}_t|\mathbf{y}_{1:T}) = p(\mathbf{x}_t|\mathbf{y}_{1:t}) \int \frac{p(\mathbf{x}_{t+1}|\mathbf{y}_{1:T})p(\mathbf{x}_{t+1}|\mathbf{x}_t)}{p(\mathbf{x}_{t+1}|\mathbf{y}_{1:t})} d\mathbf{x}_{t+1}$$

and then the following is obtained as the marginal smoothing distribution:

$$p(n_t|\mathbf{y}_{1:T}) \stackrel{\text{def}}{=} \int p(\mathbf{x}_t = [n_t\ n_{t-1}]^T|\mathbf{y}_{1:T})dn_{t-1}$$

It is apparent that the marginal smoothing distribution does not converge and the reliability of $\tilde{\mathbf{n}}_{1:T}$ is low.

The major cause is that $\mathbf{e}_{1:T}$ is assumed to be a Gaussian white noise sequence even though an analog low-pass filter has smoothed the actual measurement data. As shown in Fig. 7, the autocorrelation of $\tilde{\mathbf{e}}_{1:T} \stackrel{\text{def}}{=} Y_t - X_{\tilde{n}_t}$ is high. Therefore, to improve the model, we apply an autoregressive (AR) model to $\mathbf{e}_{1:T}$ in the next section.

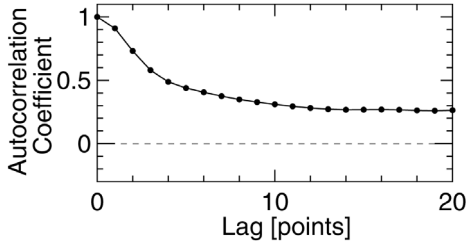

Fig. 7. Autocorrelation coefficients of a difference sequence $\tilde{\mathbf{e}}_{1:T}$ (characterizing $\tilde{\mathbf{e}}_{1:T}$ as a Gaussian white noise sequence).

3. Remodeling the Difference Sequence

3.1. Introduction of an autoregressive (AR) model

Assume that the difference component $e_t = Y_t - X_{n_t}$ is colored noise that can be described as follows using an autoregressive model:

$$Y_t - X_{n_t} = \sum_{i=1}^{k} a_i (Y_{t-i} - X_{n_{t-i}}) + w_t$$

($w_t \sim N(0, \sigma^2)$ and k is the order of the autoregressive model, which is appropriately determined). If we substitute the following for the $\sum_{t=1}^{T}(Y_t - X_{n_t'})^2$ in Eq. (1),

$$\sum_{t=1}^{T} \left[(Y_t - X_{n_t'}) - \sum_{i=1}^{k} a_i (Y_{t-i} - X_{n_{t-i}'}) \right]^2$$

and then optimize it using dynamic programming, $\mathbf{n}_{1:T}$ is obtained where $\mathbf{x}_t' \stackrel{\text{def}}{=} [n_t' n_{t-1}' \cdots n_{t-k}']^T$. This model can be transformed into a generalized state space representation by letting $y_t \stackrel{\text{def}}{=} Y_t - \sum_{i=1}^{k} a_i Y_{t-i}$ and $X_{n_t \le 0} = 0$, and the maximum likelihood of the parameters can be estimated in a similar manner as in the previous section. *LL* to be maximized to estimate the parameters is $LL(\mu_1, \mu_2, \mathbf{a}_{1:k}, \sigma^2)$. Hereafter, the estimated parameters are denoted by $(\tilde{\mu}_1, \tilde{\mu}_2, \tilde{\mathbf{a}}_{1:k}, \tilde{\sigma}^2)$.

Table 2 shows the estimated parameters when the above algorithm was applied to datasets (A) and (B), which were shown in Fig. 3 when $k = 1$ was assumed. It is apparent that by introducing the autoregressive model, the log likelihood *LL* increased significantly from 608 to 3294 and that the reliability of the estimates improved. Next, Fig. 8 shows the autocorrelation coefficients of the residual sequence $\tilde{w}_t \stackrel{\text{def}}{=} X_{\tilde{n}_t} - \tilde{a}_1 X_{\tilde{n}_{t-1}}$. This is closer to a white noise autocorrelation than in Fig. 7. However, by introducing the autoregressive model, the number of points at which $\Delta\tilde{n}_t \neq \alpha$ increased from 4 to 42 (figure is omitted). In other words, although introducing the autoregressive model increased the likelihood, an implausible $\tilde{\mathbf{n}}_{1:T}$ based on conventional knowledge was obtained.

Also, by introducing the autoregressive model, the time required for the $(\tilde{\mu}_1, \tilde{\mu}_2, \tilde{a}_1, \tilde{\sigma}^2)$ search increased. If

Table 2. Estimated parameters obtained using the actual measurement values in Fig. 3 (characterizing the differences $\tilde{\mathbf{e}}_{1:T}$ as a first-order autoregressive process)

$\tilde{\mu}_1$	$\tilde{\mu}_2$	$\tilde{a}_1$	$\tilde{\sigma}^2$	LL	$\Delta\tilde{n}_t \neq \alpha$
0.0055	0.057	0.936	0.00569	3294	42

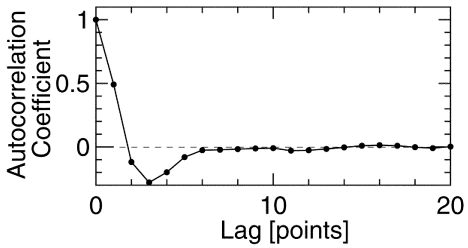

Fig. 8. Autocorrelation coefficients of a residual sequence $\tilde{\mathbf{w}}_{1:T}$ (characterizing $\tilde{\mathbf{e}}_{1:T}$ as a first-order autoregressive process).

(G_1, G_2, G_a, G_σ) denote the grid counts that were set for (μ_1, μ_2, a_1, σ^2), respectively, the required time is a $G_1 G_2 G_a G_\sigma$ multiple of the time for calculating the *LL* value once. In our environment (Pentium 4 1800 MHz), since it took approximately 100 seconds per *LL* calculation (when $T = 3201$), when all grid point counts were set to 5, introducing the autoregressive model increased the calculation time from approximately 3.5 hours to 17.4 hours. However, gauge geometry datasets are massive depending on the track length (hundreds of kilometers), and fast location adjustments are desirable.

3.2. Improvement of the estimation algorithm

As we have seen in the previous section, it is apparent that the algorithm must be improved so that an $\tilde{\mathbf{n}}_{1:T}$ that conforms to conventional knowledge is obtained and parameters having high likelihood can be estimated. Therefore, we used the fact that $(\tilde{\mathbf{a}}_{1:k}, \tilde{\sigma}^2)$ are autoregressive parameters representing the difference sequence $\tilde{e}_t$ to develop the following algorithm.

(1) Set the initial value $\mathbf{a}_{1:k}^{(0)}$ of $\mathbf{a}_{1:k}$ to 0.

(2) Maximize the log likelihood $LL(\mu_1, \mu_2, \sigma^2 | \mathbf{a}_{1:k} = \mathbf{a}_{1:k}^{(0)})$ according to a grid search to obtain provisional values of $\tilde{\mu}_1$ and $\tilde{\mu}_2$ (which are described as $\mu_1^{(0)}$ and $\mu_2^{(0)}$, respectively). In addition, use $(\mu_1^{(0)}, \mu_2^{(0)}, \mathbf{a}_{1:k}^{(0)})$ to estimate $\tilde{\mathbf{n}}_{1:T}$ according to dynamic programming.

(3) Use the Yule–Walker method to estimate $\tilde{\mathbf{a}}_{1:k}$ and $\tilde{\sigma}^2$ from the estimated difference sequence $\tilde{\mathbf{e}}_{1:k}$ (these are described as $\mathbf{a}_{1:k}^{(1)}$ and $\sigma^{2(1)}$) [5].

(4) Substitute $(\mathbf{a}_{1:k}^{(1)}, \sigma^{2(1)})$ and maximize the log likelihood *LL* again to obtain $(\mu_1^{(1)}, \mu_2^{(1)})$, and estimate $\tilde{\mathbf{n}}_{1:T}$ according to dynamic programming.

(5) Compare the newly obtained $\tilde{\mathbf{n}}_{1:T}$ with the one obtained previously and end the calculation if all entries are equal. If any entries differ, return to step (3).

(End of the algorithm)

Table 3. Transitions of parameter estimates using the actual measurements shown in Fig. 3 (characterizing differences $\tilde{\mathbf{e}}_{1:T}$ as a first-order autoregressive process and applying the alternate estimate algorithm)

	Maximum likelihood estimates of $\mu_1^{(0)}$ and $\mu_2^{(0)}$	Estimates of $\sigma^{(1)}$ and $a_1^{(1)}$ by dynamic programming	Maximum likelihood estimates of $\mu_1^{(1)}$ and $\mu_2^{(1)}$	Estimates of $\sigma^{(2)}$ and $a_1^{(2)}$ by dynamic programming
$\tilde{\mu}_1$	0.42	0.42	0.012	0.012
$\tilde{\mu}_2$	0.43	0.43	0.10	0.10
LL	608	—	3247	—
$\tilde{a}_1$	0 (fixed)	0.910	0.910	0.910
$\tilde{\sigma}^2$	—	0.00806	0.00806	0.00806
Remark				Estimation ended

Hereafter, this algorithm is termed the "alternate estimate algorithm."

Note that the $(\tilde{\mu}_1, \tilde{\mu}_2, \tilde{\mathbf{a}}_{1:k}, \tilde{\sigma}^2)$ that are obtained by the alternate estimate algorithm are not maximum likelihood estimates. $(\tilde{\mu}_1, \tilde{\mu}_2)$ are maximum likelihood estimates when $(\tilde{\mathbf{a}}_{1:k}, \tilde{\sigma}^2)$ are already known.

As a result of applying the alternate estimate algorithm to the datasets (A) and (B) shown in Fig. 3, the same $\tilde{\mathbf{n}}_{1:T}$ was obtained as before the autoregressive model was introduced. Table 3 shows transitions in estimates for the hyperparameters $(\tilde{\mu}_1, \tilde{\mu}_2, \tilde{\mathbf{a}}_{1:k}, \tilde{\sigma}^2)$. Although the log likelihood was reduced to 3247 from the 3294 of Table 2, a higher level is maintained than the 608 of Table 1. Additionally, the second graph from the bottom of Fig. 3 shows the obtained residual sequence $\tilde{w}_t \stackrel{\text{def}}{=} X_{\tilde{n}_t} - \tilde{a}_1 X_{\tilde{n}_{t-1}}$. The autocorrelation coefficients of this residual sequence are nearly the same as those in Fig. 8. The marginal smoothing distribution of this shown in Fig. 9 also converges better than the distribution in Fig. 6.

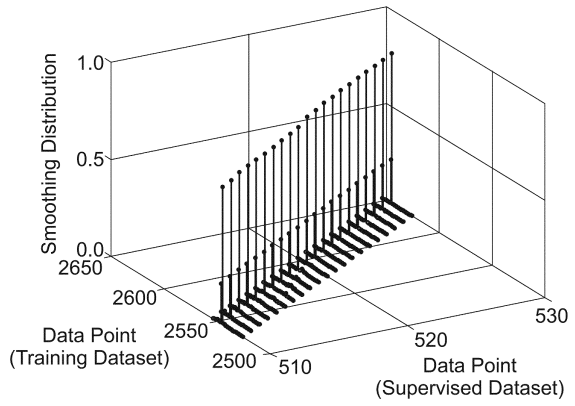

Fig. 9. Fixed-interval smoothing distribution $p(n_t|\mathbf{y}_{1:T})$ (closeup view of $t \approx 520$, modeling the differences $\tilde{\mathbf{e}}_{1:T}$ as a first-order autoregressive process and applying the alternate estimate algorithm).

Now, we discuss the required time for the grid search. If the alternate estimate algorithm is repeated M times, the sum total of the required times is $100[G_1G_2G_\alpha + (M - 1)G_1G_2]$. Since $M = 2$ for the calculations in Table 3, when all grid point counts were set to 5, the calculation time was reduced to approximately 4.2 hours from 17.4 hours. Note that M changed from 2 to approximately 5.

Therefore, the alternate estimate algorithm seems practical for obtaining $\tilde{\mathbf{n}}_{1:T}$ from the target data.

4. Discussion

To verify the alternate estimate algorithm, we performed the following simulation.

(1) Establish fictional gauge geometry and slipping or sliding to create data sequences for the supervised and training datasets (before the interpolation). Let $\mathbf{n}^*_{1:T}$ denote true values of the data points.

(2) Independently create two white noise sequences as realization from the same normal distribution to simulate measurement noise and add them respectively to the two data sequences that were created in step (1).

(3) Create $\mathbf{Z}_{1:T}$, the sum of the supervised gauge geometry and the noise. Then sequentially calculate $Y_1 = Z_1$, $Y_t = Z_t + a^* Y_{t-1}$ $(2 \le t \le T)$ to create the supervised dataset $\mathbf{Y}_{1:T}$. Also create the training dataset in a similar manner. (These sequences are shown in the top two graphs in Fig. 10.) Note that in this paper, $a^* = 0.8$.

(4) Estimate parameters $(\tilde{\mu}_1, \tilde{\mu}_2, \tilde{a}_1, \tilde{\sigma}^2)$ and use those parameters to estimate $\tilde{\mathbf{n}}_{1:T}$.

(End)

The two results, obtained by the maximum likelihood estimation and by alternate estimate algorithm (see Table 4), are equal, and, therefore, the $\tilde{\mathbf{n}}_{1:T}$ were also both the same. In addition, $\tilde{a}_1$ and $\tilde{\sigma}^2$ are both close to the true values.

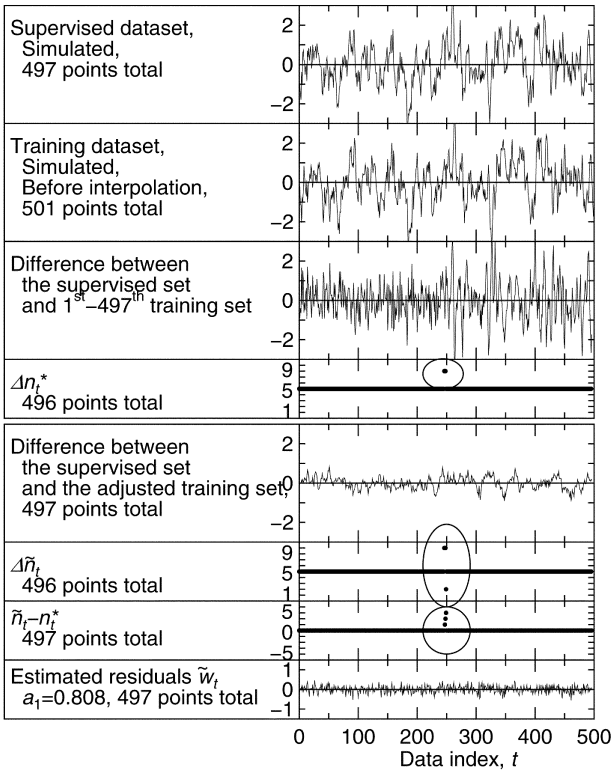

Fig. 10. The result of adjusting sampling locations of the simulated datasets.

Table 4. Parameter estimation results for the simulation in Fig. 10

	Maximum likelihood estimation	Alternate estimate algorithm	True value
$\tilde{\mu}_1$	0.	0.	—
$\tilde{\mu}_2$	0.42	0.42	—
$\tilde{a}_1$	0.808	0.808	0.8
$\tilde{\sigma}^2$	0.0299	0.0299	0.0286
LL	147	147	—

We believe that this is because $\mathbf{e}_{1:T}$ has been modeled correctly. However, as Fig. 10 shows, $\tilde{\mathbf{n}}_{1:T} \neq \mathbf{n}^*_{1:T}$.

The results of the parameter estimations using actual measurement data differed for maximum likelihood estimation and the alternate estimate algorithm as shown in Tables 2 and 3. We believe that this is because $\mathbf{e}_{1:T}$ cannot be described accurately by a first-order autoregressive model for the following reasons. That is, the main causes are that the measurement data contains irregular noise (for example, in the neighborhood of $t = 2300$) as is apparent from Fig. 3 and the filter that is actually applied is an analog low-pass filter that varies with time.

5. Conclusions

The sampling locations of two measurement datasets of a track geometry obtained with different runs of a track inspection car can be approximately aligned by dynamic programming if we model the wheel rotation and location detection pulse. Also, using the fact that the optimal solution according to dynamic programming is a MAP estimate in the Bayesian framework enables unknown parameters included in the model to be estimated by using the maximum likelihood method. When a low-pass filter is employed to represent the measurement datasets, its effect can be reduced if an autoregressive (AR) model is applied to the difference sequence.

REFERENCES

1. Ibaraki T. Dou-teki Keikaku-hou, RisanSaitekika-hou to Arugorizumu. Iwanami Shoten; 1993. p 61–67. (in Japanese)
2. Akaike H. Likelihood and the Bayes procedure. In Bernardo JM, DeGroot MH, Lindley DV, Smith AFM (editors). Bayesian Statistics. University Press; 1980. p 143–166.
3. Godsill S, Doucet A, West M. Maximum a posteriori sequence estimation using Monte Carlo particle filters. Ann Inst Statist Math 2001;53:82–96.
4. Kitagawa G. Jikeiretsu-moderu no Yuudo-keisan to Parameta-suitei, FORTRAN77 Jikeiretsu-kaiseki Puroguramingu. Iwanami Shoten; 1993. p 214–217. (in Japanese)
5. Ishiguro M. Jiko-kaiki moderu no Atehame, Jikeiretsu-kaiseki no Houhou. Asakura Shoten; 1998. p 64–70. (in Japanese)

AUTHORS (from left to right)

Masako Kamiyama completed her master's course in 1992 with a specialty in Resource Engineering at Waseda University. Currently, she is employed in the Track Technology Division of the Railway Technical Research Institute, and she completed her doctoral course in 2004 in the Department of Statistical Science at the Graduate University for Advanced Studies (Sokendai). She is a member of the Japan Society for Industrial and Applied Mathematics and Japan Statistical Society.

Tomoyuki Higuchi (member) completed his doctoral course in the Department of Geophysics, in 1989 at the University of Tokyo. He holds a Ph.D. degree in Science. Currently, he is a professor in the Department of Statistical Modeling and Vice Director-General at the Institute of Statistical Mathematics, Research Organization of Information and Systems. His specialty is statistical analysis, in particular, Bayesian modeling for knowledge discovery. He is a member of the Japan Statistical Society, the Society of Geomagnetism and Earth, Planetary and Space Sciences, the American Geophysical Union, and the American Statistical Association.