UNIWERSYTET TECHNOLOGICZNO-PRZYRODNICZY
IM. JANA I JĘDRZEJA ŚNIADECKICH W BYDGOSZCZY
ZESZYTY NAUKOWE NR 258
TELEKOMUNKACJA I ELEKTRONIKA 14 (2011), 69-78

SOLUTION IMPLEMENTATION BASED ON MODIFIED KALMAN FILTER FOR PURPOSE OF BUS ARRIVAL TIME PREDICTION

Damian Ledziński[1], Michał Jezierski[2], Beata Marciniak[1], Tomasz Marciniak[1]

[1] Institute of Telecommunications,
Faculty of Telecommunications and Electrical Engineering
University of Technology and Life Sciences
Al. S. Kaliskiego 7, 85-789 Bydgoszcz, Poland
[Damian.Ledzinski, BeataMarciniak,Tomasz.Marciniak]@utp.edu.pl

[2] Electronics Company SIMS, Bydgoszcz
ambinanitelo@gmail.com

Summary : This paper describes use of Kalman's filter for prediction of time of arrival of bus. Kalman filter is recursive algorithm determining the minimum-variance estimate of the state vector of dynamic system, based on the measurement of inputs and outputs of the system. Three prediction algorithms used: difference algorithm, traditional Kalman filter and Kalman filter with changing weights of input data. Authors studied the bus arrival time predictions. Used for this purpose data send by radio from vehicles to prediction server. The smallest average prediction error obtained for the Kalman filter with variable weights.

Keywords: Kalman filter, time prediction

1. INTRODUCTION

In the era of technological development especially in large cities, is increasing need for effective and fast movement. Very often, public transportation is used for this purpose. Thus, there can be observed an increasing demand for accurate information on the transportation possibilities. The traditional schedule fixed in advance is not sufficient any more, travelers expect information in real time, along with the changing traffic conditions in the city.

The possibilities, of presenting such data are also increasing, including electronic boards at bus stops, offices and shopping centers and dynamically updated web pages being browsed by the mobile phones. However, there is a problem of storage data on the vehicle position and its processing in order to be able to predict traffic delays.

The bus arrival time prediction problem already was discussed in the literature. Usefulness of different types of input information and computing methods has been studied [1,2,3,4,5,6,7,8,9,10]. In this paper the authors present a system which public transportation vehicles, send the information about the bus arrival at the next stops via radio. This information is collected and processed by the prediction server (Fig. 1). The authors present and compare three prediction algorithms: difference algorithm, Kalman filter and Kalman filter with changing weights of input data. The Kalman filter algorithm has been used in many subjects including: weather prediction [11], asset pricing [12], navigation [13] and visual tracking [14].

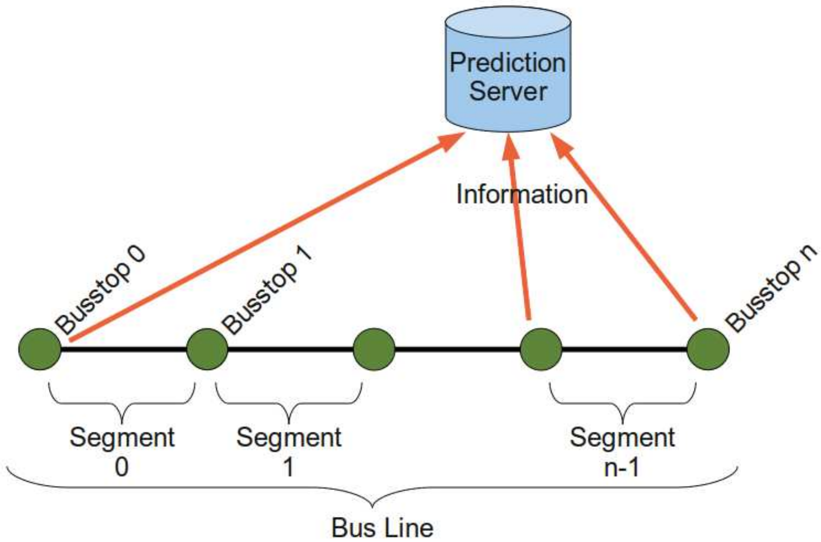

Fig. 1. Bus line schematic

Each bus line can be described by a graph where a stops are nodes and segments are the edges. If each distance is assigned with the time in which is defeated, we receive a weighted graph. The sum of the lengths of these edges describes the travel time for whole route. Theoretical travel time is regulated by schedule.

1.1. DIFFERENCE ALGORYTHM

The simplest to be used is the differential algorithm. It checks delay of bus at the stops, defined as the difference between real and scheduled time of departure from the bus stop.

$$\Delta t(B_k) = t_R(B_k) - t_s(B_k) \tag{1}$$

where:

$\Delta t(B_k)$ – delay on bus stop B_k,

$t_R(B_k)$ – real time of departure from the bus stop B_k,

$t_s(B_k)$ – scheduled time of departure from the bus stop B_k.

The basic idea of the difference algorithm is that, if the vehicle is late at bus stop B_k, by the time Δt, then at the next stop B_{k+1} it is also a late by the time Δt

$$\Delta t(B_{k+1}) = \Delta t(B_k) \tag{2}$$

Estimated time of arrival of the bus at the next stop, is the scheduled time corrected by the delay from the previous stop

$$t_x(B_{k+1}) = t_S(B_{k+1}) + \Delta t(B_k) \tag{3}$$

where $t_x(B_{k+1})$ is estimated arrival time from bus stop B_{k+1}, so

$$t_x(B_{k+1}) = t_S(B_{k+1}) + t_R(B_k) - t_S(B_k) \tag{4}$$

Difference algorithm was used here as a reference point for determination of efficiency of the algorithms presented further in this paper.

2. RESEARCH METHODOLOGY

For all the algorithms presented in this paper test method was the same. Data for the tests was taken from an existing system. Those were the times of departures from the bus stops one of the most frequented used lines in one of the Polish cities. These times were collected one month. Totally, there were more than 50,000 departures.

For the study, the authors developed a simulator, implemented in Java programming language (see Fig. 5). It introduced the data to be used for computing of prediction (the real times of departure from previous stops, or previous rides of the tested distance). Then, as result calculated the difference between predicted time and real travel time (prediction error).

2.1. INTRODUCTION TO THE KALMAN FILTER

The considered model consists of two linear stochastic difference equations:

$$x_n = x_{n-1} + w_{n-1} \tag{5}$$

$$y_n = x_n + v_n \tag{6}$$

Our aim is to estimate unknown (unobservable) variable x, whose evolution in time is described by the first equation (5). Quantity x is assumed to be 'true' value and unknown to us. All of the values $y_0,\ldots,y_n$ are known and indicate measurements of x in discrete points of time $0,\ldots,n$. Because of uncertainty of measurement process, the second equation (6) contains v_n summand. We assume here, that in each time step, ie. for each n, process noise w_n and measurement noise v_n are both white and Gaussian (v_n and w_n are uncorrelated random variables)

$$w_n \sim N(0,Q) \tag{7}$$

$$v_n \sim N(0,R) \tag{8}$$

In the following paper we considered the filter to be stationary, which means that variances Q and R are fixed. We will use the following notation:

$\hat{x}_n^-$ – a priori estimation taken in step n with $n-1$ given observations, ie. $\hat{x}_n^- = \varepsilon\left[x_n \mid y_0, \ldots, y_{n-1}\right]$,

$\hat{x}_n$ – a posteriori estimation – process state estimation with n given observations, ie. $\hat{x}_k = \varepsilon\left[x_n \mid y_0, \ldots, y_{n-1}, y_n\right]$,

$e_n^- = x_n - \hat{x}_n^-$ – a priori estimation error,

$e_n = x_n - \hat{x}_n$ – a posteriori estimation error.

Kalman filter is a tool used to find an optimal estimate of x which minimizes the a posteriori estimation error P_n, ie. minimizes mean-squared error given by the formula

$$P_n = \varepsilon\left[e_n^2\right] = \varepsilon\left[\left(x_n - \hat{x}_n\right)^2\right]. \tag{9}$$

Under above assumptions, estimation $\hat{x}_n$ of system state x_n which minimizes mean-squared error (9), is given by linear combination

$$\hat{x}_n = \hat{x}_n^- + K_n\left(y_n - \hat{x}_n^-\right), \tag{10}$$

where

$$K_n = \frac{P_n^-}{P_n^- + R}, \tag{11}$$

$$P_n = \left(1 - K_n\right)P_n^-, \tag{12}$$

$$P_n^- = P_{n-1} + Q, \tag{13}$$

$$\hat{x}_n^- = \hat{x}_{n-1}. \tag{14}$$

Reader interested in detailed proof of above equations may refer to original R. Kalman's article [15]. A theoretical introduction to Kalman filter theory can be found in many papers including [16,17,18].

Roughly speaking, Kalman filter is the way to compute an optimal (ie. with minimal P_n) estimate from two given values. It is a kind of weighted mean which uses process and measurement uncertainties Q and R instead of standard weights. In our case for every n, Kalman filter algorithm estimates $\hat{x}_n$, basing on $\hat{x}_n^-$ and y_n (see Fig. 2). From equation (10) it follows, that when Q is significantly greater than R then $\hat{x}_n \approx y_n$. Analogously, if R is significantly greater than Q, then $\hat{x}_n \approx \hat{x}_n^-$.

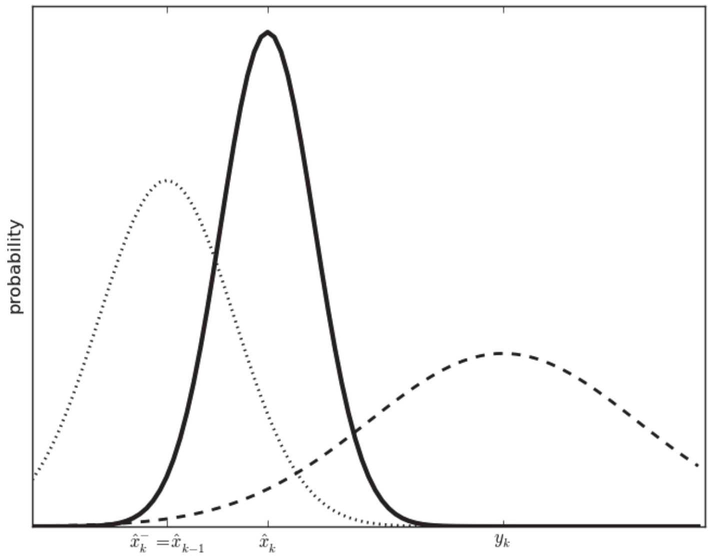

Fig. 2. Estimation taken in k-th step under assumption $Q < R$.

Due to the recursive nature of the estimation algorithm, the process can be organized into two stages: time update step (a priori estimation) and the measurement update step (a posteriori estimation). The above two-step approach is shown in the following diagram (Fig. 3), where left side presents a posteriori estimation, and right side a priori estimation (indices n are omitted for simplicity)

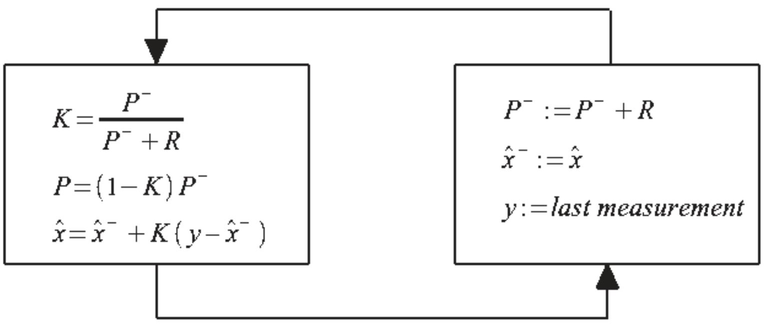

Fig. 3. Time update and measurement update steps

2.2. USAGE OF THE KALMAN FILTER

In order to improve efficiency (minimize prediction error), we decided to use the Kalman filter. The input data of filter were previous travel times on tested distance. That is, travel times of this distance immediately before the tested one.

In each distance the server uses an independent filter. When the vehicle has traveled the distance, the filter performs the next iteration. This iteration was calculated from the new input data. The result of this iteration was the prediction of travel time the next vehicle on the same distance. Moreover, a filter at each iteration did not receive a single travel time, it received an average weighted from the last three times of traveling over a given distance.

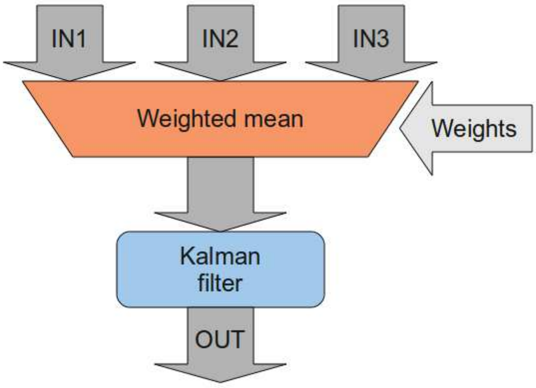

Figure 4 Filter construction

$$\Delta t_x(V_m) := F\left(\frac{\sum_{i=1}^{3} w_i \cdot t_D(V_{m-i})}{\sum_{i=1}^{3} w_i}\right) \tag{15}$$

where:

- $F(x)$ – The Kalman filter iteration with input data x,
- $\Delta t_x(V_m)$ – estimated segment traveling time by vehicle V_m,
- w_i – weight for i-th parameter,
- $t_D(V_{m-i})$ – travel time over a distance by earlier vehicle.

Looking for the optimal solution, assumed weight was 1, 0.6, 0.3.

2.3. KALMAN FILTER WITH CHANGING WEIGHTS

To improve results, we tested the previous algorithm with different values of weights w_1, w_2, w_3. Our goal was to find the optimal values of weights. The best weights were those with the lowest prediction error. Such a test was performed for each distance separately.

```
private final double Q, R, M, F;
private double p = 1.0;
private double xhat = 1.0;

public double predict(double inputValue) {
  double x_hat_k_minus;
  double p_k_minus = Math.pow(F, 2) p + Q;
  double K = p_k_minus * M / (p_k_minus * Math.pow(M, 2) + R);
  double x_hat_k = x_hat_k_minus+ K*inputValue-M*x_hat_k_minus);
  double p k = (1 - K) * p k minus;
  p = p k;
  xhat = x_hat_k;
  return x_hat_k;
}
```

Fig. 5. Implementation of the Kalman filter in Java programing language

Results (most optimum weight for each section) are shown in Table 1. Notice that there are different values for each distance. Inspired by this observation, we concluded that weight could be periodically modified by the simulations were carried out on historical data from some period.

Table 1. Optimal values of weights on each segment

Distance	w_1	w_2	w_3
0	1	0,7	0,7
1	1	1	0,8
2	1	0,4	0,4
3	1	0,4	0,4
4	1	1	1
5	1	0,4	0,4
6	1	1	1
7	1	1	1
8	1	0,6	0,6
9	1	0,1	0,1
10	1	0,8	0,7
11	1	0,8	0,1
12	1	1	1
13	1	0,5	0,5
average	**1**	**0,69**	**0,62**

3. RESULTS

Applying discussed in the previous chapter the methods of determining the time prediction obtained the following results (table 2). For better clarity, derived calculations have been rounded to 2 decimal places. The smallest average prediction error was obtained for the Kalman filter with variable weights. Some individual time prediction errors received for the differential algorithm had little value (within 0 and 1 in Table 2). However, in this algorithm all errors are cumulative, making it impossible to obtain satisfactory results.

Table 2. The average prediction error for each distance for different algorithms

Distance	Diference Algorithm	Kalman Filter	Kalman Filter with changing weights
0	7,61	19,63	5,22
1	7,63	15,65	6,74
2	16,27	15,03	14,29
3	59,15	13,10	10,56
4	17,51	34,24	18,80
5	24,85	20,20	24,67
6	55,97	27,63	7,42
7	35,36	15,75	20,09
8	40,06	59,51	40,33
9	59,63	19,78	50,79
10	18,84	40,36	11,65
11	37,97	45,01	33,44
12	38,64	31,77	26,70
13	33,77	23,64	19,55
average	**32,38**	**27,24**	**20,73**

4. CONCLUSIONS

As shown in the table 2, the Kalman filter with variable weights has the best effectiveness. Kalman filter with fixed weights produces better results than a difference algorithm.

The main advantages of the difference algorithm are its relatively good performance and simplicity. The disadvantage is that it requires constant predefined schedule. It cannot be used in situations when we do not have a regular schedule. If the schedule is poorly prepared, that is , the travel times of the distances are poorly matched, the algorithm will produce more mistakes. Delayed drivers will probably try to make up for the delay, whereas, the difference algorithm assumes the delay to be constant.

The advantage of using a Kalman filter is better efficiency, especially with many vehicles on the line. If at any point in the city there is a traffic jam, the filter quickly adopts to the situation (fast response to changing time of travel through a given distance). In addition, it does not require a fixed schedule. This filter will not work correctly when the line is loaded less than other bus line (eg, a few rides during the day).The filter requires historical data. So, when the program starts and there is no data collected, will not work properly. To make this possible, it passes several iterations and it can takes whole day.

Kalman filter with changing weights proved to be the best although it needed performance of periodic simulations for different values of weights, which requires much computing power. And it is its biggest disadvantage. Moreover, it requires storage of lots of historical data.

The described method can be applied in other areas of technology, wherever we can use the Kalman filter and collect historical data.

In the future, we will test the Kalman filter having more parameters than just the last tree travel times. Perhaps, expected results would be given by a simulation using the optional values of fixed parameters of Kalman filter.

BIBLIOGRAPHY

[1] Kajan E., 2002. Information technology encyclopedia and acronyms. Springer, Berlin Heidelberg New York.

[2] Broy M., 2002. Software engineering – From auxiliary to key technologies. In: Broy M., Denert E. (eds). Software Pioneers. Springer, Berlin Heidelberg New York.

[3] Che M, Grellmann W, Seidler S, 1997, Appl Polym Sci 64:1079-1090.

[4] Ross D.W., 1977. Lysosomes and storage diseases. MA Thesis, Columbia University, New York.

[5] Padmanaban, R.P.S., Divakar, K., Vanajakshi, L., Subramanian, S.C., 2010. Development of a real-time bus arrival prediction system for Indian traffic conditions. In: Intelligent Transport Systems, IET. vol. 4, issue 3, pp. 189-200.

[6] Hao Chu, Yun Cai, Xiaoguang Yang, 2007. Research on Bus Arrival Time Prediction Based on Multi-Source Traffic Information. In: Telecommunications, ITST '07. 7th International Conference on ITS, pp. 1-5.

[7] Jian Zhang, Ling Yan, Yin Han, Jing-Jing Zhang, 2000. Study on the Prediction Model of Bus Arrival Time. In: Management and Service Science,. MASS '09. pp. 1-3.

[8] Jeong, R., Rilett, R., 2004. Bus arrival time prediction using artificial neural network model. In: Intelligent Transportation Systems,. Proceedings. The 7th International IEEE Conference, pp. 988-993.

[9] Latos P., Dubalski B., Marciniak T., Marciniak B., 2010. Demand forecasting for spare parts. Zesz. Nauk. UTP w Bydgoszczy, Telekomunikacja i Elektronika 13, pp. 103-114.

[10] Padmanaban, R.P.S., Vanajakshi, L., Subramanian, S.C., 2009. Automated Delay Identification for Bus Travel Time Prediction towards APTS Applications, Emerging Trends in Engineering and Technology (ICETET), 2nd International Conference, pp. 564 569.

[11] Galanis G., Louka P., Katsafados P., Kallos G., Pytharoulis I., 2006. Applications of Kalman filters based on non-linear functions to numerical weather predictions, Ann. Geophys., vol. 24, pp. 1-10.

[12] Bidarkota P., Dupoyet B., 2006. Asset Pricing with Incomplete Information In a Discrete Time Pure Exchange Economy; Florida International University, Department of Economics.

[13] Brock L.D., Schmidt G.T., 1970. General Questions on Kalman Filtering in Navigation Systems, Chapter 10 of Theory and Applications of Kalman Filtering C.T. Leondes, Editor, NATO AGARD.

[14] Funk, N., 2003. A Study of the Kalman Filter applied to Visual Tracking; University of Alberta.

[15] Kalman R.E., 1960. A New Approach to Linear Filtering and Prediction Problems, Transaction of the ASME – Journal of Basic Engineering, pp. 35-45.

[16] Jacobs O.L.R., 1993. Introduction to Control Theory; 2nd Edition. Oxford University Press.
[17] Welch G., Bishop G., 2006. An Introduction to the Kalman Filter; Department of Computer Science at the University of North Carolina at Chapel Hill Tech.
[18] Maybeck, Peter S., 1979. Stochastic Models, Estimation, and Control, vol. 1; Academic Press Inc.

IMPLEMENTACJA FILTRU KALMANA DO PROGNOZOWANIA CZASU PRZYBYCIA AUTOBUSÓW

Streszczenie

W pracy przedstawiono zastosowanie filtru Kalmana do prognozowania czasu przybycia autobusów. Filtr Klamana to algorytm rekurencyjnego wyznaczania minimalnowariancyjnej estymaty wektora stanu układu dynamicznego, na podstawie pomiaru wejść i wyjść tego układu. Zbadano trzy algorytmy predykcji: algorytm różnicowy, tradycyjny filtr Kalmana oraz filtr Kalmana ze zmiennymi współczynnikami. Autorzy badali odchylenie od prognozowanego czasu przyjazdu autobusów. Używano do tego celu danych przesyłanych drogą radiową z autobusów do serwera predykcji. Najlepsze wyniki uzyskano dla filtru Kalmana ze zmiennymi współczynnikami.

Słowa Kluczowe: filtr Kalmana, prognozowanie czasu