ACTA IMEKO
ISSN: 2221-870X
April 2017, Volume 6, Number 1, 13-19

Hybrid backward simulator for determining causal heater state with resolution improvement of measured temperature data through model conformation

Yukio Hiranaka, Shinichi Miura and Toshihiro Taketa

Yamagata University, Jonan 4-3-16, Yonezawa, 992-8510 Yamagata, Japan

ABSTRACT
We are developing a backward simulator, which determines the unknown system input from the system output by using a system model. However, its processing time would increase enormously if the simulation model requires the multiple case branching, which is typical for backward simulations. In some target applications, we can use forward simulation processing in the backward data flow with significant reduction of processing time. This paper shows an example of such application to determine system input of heater operation from measured data of room temperature. Although the resolution of measurement restricts the performance of the simulation result, we also used the model to improve the resolution of measured data and show its effect to simulation. Furthermore, we show the result of reduction of noise caused by quantizing LSB jitters.

Section: RESEARCH PAPER

Keywords: backward simulation; temperature measurement; inverse problem; precision improvement; LSB noise reduction

Citation: Yukio Hiranaka, Shinichi Miura and Toshihiro Taketa, Hybrid backward simulator for determining causal heater state with resolution improvement of measured temperature data through model conformation, Acta IMEKO, vol. 6, no. 1, article 3, April 2017, identifier: IMEKO-ACTA-06 (2017)-01-03

Section Editor: Paul Regtien, The Netherlands

Received May 28, 2016; **In final form** March 14, 2017; **Published** April 2017

Funding: This work was supported by JSPS KAKENHI Grant Number 255400006

Corresponding author: Yukio Hiranaka, e-mail: zioi@yz.yamagata-u.ac.jp

1. INTRODUCTION

To estimate heater operation from the data of room temperature change is a typical inverse problem [1], [2]. And it is a kind of ill-conditioned problem because a slight error in the data would extremely disturb the results [3]. However, such a problem is common for measurement and system diagnostics, and an important part of measurement.

If the relation between input and output is linear, it is a problem of deconvolution and there are many studies including super-resolution [4]. Subtractive deconvolution [5] may be applied if available data are impulsive. Unfortunately, temperature change is a long trailing phenomenon and is very sensitive to measurement resolution and error [3]. Our idea to tackle this is to consider measurement resolution inevitable and to treat it explicitly by defining a range for each input or output signal.

There are two ways to search deconvolution results for range signal models. One is a try and modify method to test some signal within a certain divided range and to judge whether it matches the given output data. It is a forward simulation because it requires many trials, regardless of whether a convergence method is used or not. The other is an inference method starting from the given output data to the input in backward direction. It is a normal method for problem solving, but it is not always possible to find a good solution. However, we may perform a backward simulation in a similar manner to the forward simulation, starting with a range divided output signal, if we can make a backward processing model.

There are backward simulation applications in many fields such as process scheduling [6], initial position estimation of physical objects [7], and software debugging [8].

There exist many difficulties in creating a backward system model. A typical backward simulation requires case branching when it has multiple possible conditions on its backward trace. However, we have experiences to cope with them, and have shown that some physical restriction may effectively reduce the number of case branches [9], [10]. We have been focusing our attention on utilizing temporal model and strict facts such as nonnegative property and causality that any current value must not affect past values. Under such conditions, we can perform a backward simulation effectively.

In this paper, we show two methods for efficiently performing backward simulation and for effectively improving measurement resolution. The first method is for creating a backward simulator by incorporating a forward simulation model. It extremely suppresses case-branching processing in the example of this paper. The second method is for improving resolution of the given measured data by applying a strict model of the system.

Hereinafter, we describe the concept of backward simulation and target simulation model in Section 1, hybrid implementation of backward integration loop in Section 2, measured temperature data and estimation of model parameters in Section 3, simulation results for error free temperature data in Section 4, effect of quantization error and countermeasure to it in Section 5, simulation results for real measurement data in Section 6, discussion in Section 7, and conclusion in Section 8.

2. BACKWARD SIMULATION AND MODELS

Here, we explain the concept of backward simulation with an example application of inference of heater activity. Figure 1 shows the structure of our forward simulation model for room temperature change caused by an electrical heater. Black dots denote branching points, circles with plus sign denote summation points, and arrows denote the direction of physical information flow. A block with z^{-1} means one sampling time delay, which is a member of the integration loops. We adopted two integration loops because the room temperature increases even after the time the heater is turned off. The first integration loop corresponds to the neighbor of the heater that accumulates heat locally, and the second loop corresponds to the whole room heat reservoir. The parameters *A*, *B* are decay factors due to heat transfer, *C* is the specific heat of the room which converts heat to temperature.

Figure 1 can be expressed by the following equation,

$$\dot{q}_1 = \begin{cases} w - Aq_1 & (0 \le t < t_i) \\ -Aq_1 & (t_i \le t) \end{cases}, \qquad (1)$$

$$\dot{q}_2 = Aq_1 - Bq_2$$

where q_1 and q_2 are accumulated heat in the first loop and the second loop, t is time, and the heater of w (W) is on for 0 to t_i. We can obtain q_2 as

$$q_2 = \begin{cases} w\left\{B(1-e^{-At}) + A(1-e^{-Bt})\right\}/B(B-A) & (0 \le t < t_i) \\ w\left\{B(e^{At_i}-1)e^{-At} - A(e^{Bt_i}-1)e^{-Bt}\right\}/B(B-A). & (t_i \le t) \end{cases} \qquad (2)$$

Figure 2 shows the implemented software objects and connection diagram for the simulation of Figure 1. Each object has a corresponding function, “heater” for heating for a specified time duration, “wsum” for summing heat with wattage input port, “csum” for summing heat, “br” for branching with the same output values, “dly” for one sampling time delay, “co” for coefficient multiplier, and “temp” for temperature recording. These objects are coded in Scala classes, which use a Java virtual machine, and dly1 and dly2 are instance objects of the class dly, for example. All the objects have independent GUI windows which accept local settings and display status and parameters of each object.

Data flow on the connection links consist of time and values. As an example, heating power 800 W from “heater” at its output port “o” at the time of 10.0 s is shown in a XML style UCF (universal communication format) message [9], [10] <sim><s>heater<s>o</s></s><t>10.0</t>800.0</sim>, where sim is the simulation controller which redirects this message from “o” port of “heater” to “i” port of “sum1” as specified in the simulator’s connection table [10]. The <s> tag indicates the source of the message. The source tag is nested in this case to indicate the port “o” of “heater.”.

In the backward simulation (Figure 3), UCF messages flow in the backward direction indicated by dashed lines with reversed arrows. The node “temp” is the starting object which sends the time sequenced temperature in the reversed order, one pair of time and temperature data in each UCF message. The temperature in the backward simulation is expressed by a value range which expresses the minimum and the maximum temperature as “0.0, 10.0”, which means that the temperature is in the range from 0.0 inclusive to 10.0 exclusive [10].

By narrowing the range, we can control the resolution of the simulation. Simulation parameter “ndiv” set by the starting block “temp” specifies the number of divisions. As an example, when the whole range is from 0.0 to 10.0 and ndiv is equal to 10, the value 4.5 is expressed in the UCF message as “4.0, 5.0” as the divided range width is 1.0.

3. HYBRID SIMULATOR AND IMPLEMENTATION

Two important features of the simulator are time synchronization and hybrid backward simulation. Figure 4(a)

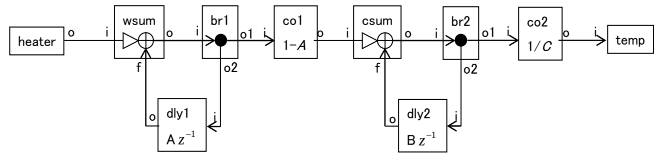

Figure 2. Forward simulation objects and connections corresponding to Figure 1.

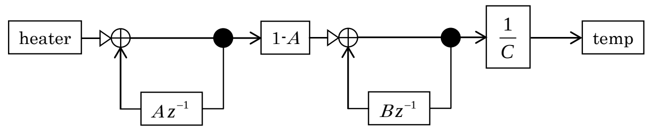

Figure 1. Forward simulation model of room temperature.

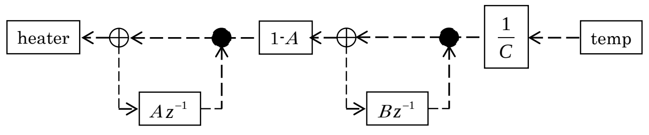

Figure 3. Backward simulation model corresponding to Figure 1.

simplifies a forward integration loop in Figure 1. The summation node must gather the same time data from the two incoming links, the input x(i) and the feedback f(i)=y(i-1) to form the output y(i)=y(i-1)+x(i).

A fully synchronized simulator will do the work naturally. However, we intend to perform our simulation in a distributed processing environment, and we prefer asynchronous processing as long as it is possible. Then, the summation node is designed to have an input record table which keeps time and value pairs received at the left input port (triangle arrowed port in Figure 1 up to Figure 13). The data message from the feedback port (from the delay node) triggers the summation process by picking up the data of the same time from the input record table.

Figure 4(b) shows the backward version of Figure 4(a). We need to calculate two backward outputs from a single backward input at the summing point, satisfying y(i)=x(i)+f(i). The computation intensive solution is to simulate all the pairs of x(i) and f(i), matching the equation. A practical solution is to perform the simulation for a finite number of divided range pairs, e.g. {x,f} pairs of {0.0 to 2.5, 2.5 to 5.0} and {2.5 to 5.0, 0.0 to 2.5} to match the output of 4.5 [10]. A case branching mechanism is needed for such processing.

However, if we apply another solution which uses a forward simulation object in the backward simulation as in Figure 5, we can avoid the expansion of processing time caused by the case branching. In Figure 5, we can formulate the process as x(i)=y(i)–f(i) and f(i)=y(i+1). The branch node sends received backward data in time reverse order through the two links to the sum node and to the delay node. The same input record table used in Figure 4 can be utilized to keep y(i) required by the reverse delayed feedback signal of f(i)=y(i+1).

We have to describe the detailed processing of the sum node. In the backward simulation, data flows are expressed as a range (the minimum and the maximum). Then, the backward calculation must handle the range information. If the range from the right is (a, b), which means that the minimum value is a and the maximum value is b, and the range from the bottom is (c, d), the output through the left port should be calculated as (a-d, b-c) to cover the broadest value range. However, x(i) must be positive or zero as it expresses heat. If a-d is less than zero, it must be substituted by zero. Furthermore, if b-c is less than zero, the case simulated is not a feasible one.

Figure 6 shows the resulting practical hybrid backward simulator, and Figure 7 shows implemented objects and connections. The simulation objects in Figure 7 are the same objects as in Figure 2, with backward processing capability except dly. The objects “co1” and “co2” divide their incoming backward data by their coefficients, “br1” and “br2” pass through their incoming backward data to their two backward outputs. The simulator in Figure 7 needs three consecutive backward inputs to start as the first data stops at csum and the second data stops at wsum because there will be no matching time data coming from the other backward port.

We describe here the mechanism of model mismatch detection in detail. Sum objects (wsum and csum) in Figure 7 have two arriving inputs in the backward simulation. We note the backward input from the right of csum at the time sample of t as $v(t)$, the other backward input can be expressed as $Bv(t-1)$ because dly2 node delays the backward flow signal for one sample time and multiplied by a constant coefficient *B*. The backward output $u(t)$ from csum can be expressed as equation (3) and must be positive or zero because it expresses heat.

$$u(t) = v(t) - Bv(t-1) \geq 0. \tag{3}$$

The backward output from wsum can be expressed as the following equation,

$$\begin{aligned}\frac{u(t)-Au(t-1)}{1-A} &= \frac{v(t)-Bv(t-1)-A\{v(t-1)-Bv(t-2)\}}{1-A} \\ &= \frac{v(t)+ABv(t-2)-(A+B)v(t-1)}{1-A} \geq 0.\end{aligned} \tag{4}$$

If the both inequalities are not satisfied for the maximum value of the range at any time point, the backward simulation is failed and the starting condition must not occur for the simulation model.

4. REAL TEMPERATURE CHANGE AND MODEL SIMULATION

At first, we verified the correctness of the simulator. Figure 8 shows the result of a room temperature measurement when the infrared heater (800 W) on the floor was turned on for the duration from 0 s to 180 s in a tiny room of 3.6 m^3. The sensor is a Sensirion’s SHT71, which has 0.01 degree resolution, placed near the heater at 1 m high in the room.

Figure 8 also shows a simulated temperature change using the parameters *A* (0.095/10s), *B* (=0.9938/10s) and *C* (24200 cal/deg). To determine these parameters, we assume that the heat loss rate of the room (*B*) is less than that of the heater appliance (*A*). Then the temperature decline after the

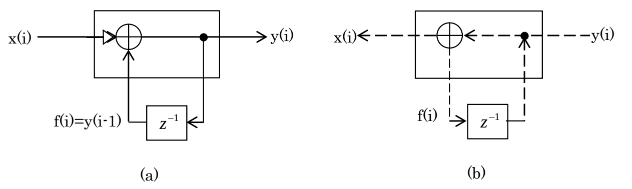

Figure 4. Forward integration loop (a) and backward integration loop (b).

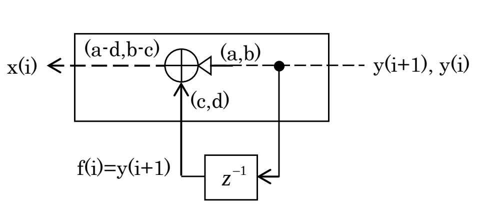

Figure 5. Hybrid backward integration loop.

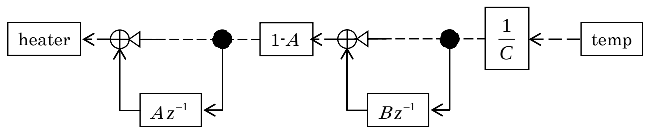

Figure 6. Hybrid backward simulation model.

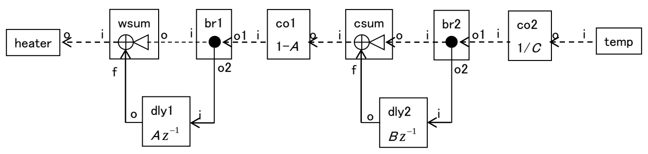

Figure 7. Hybrid backward simulation objects and their ports.

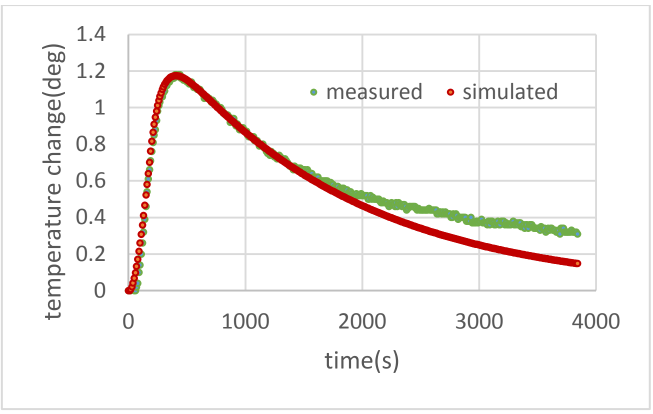

Figure 8. Measured and simulated temperature change caused by heating for 180 s from the start.

temperature peak is estimated by the following equation,

$$\log q_2 = -Bt + const. \quad (t_i \le t). \tag{5}$$

We define t_p as the peak temperature time and estimate A by the following equation derived from equation (2) ,

$$e^{(A-B)t_p} = (e^{At_i} - 1)/(e^{Bt_i} - 1). \tag{6}$$

The heat capacity parameter C is estimated as the magnification factor for fitting the measured data with the simulated temperature data which was calculated by using the simulator's forward function with the estimated parameters A and B. Those parameters should be fine-tuned to closely fit the measured data. However, the rear part of the curve in Figure 8 cannot be fit by our model. We may need the third heat accumulation loop for representing wall temperature change and heat release to the environment. As will be shown in Section 6, the above approximation was almost enough because large measured values around the peak are matched.

Figures 9 and 10 are the results of the backward simulation when the simulated temperature change in Figure 8 was fed backwardly from "temp" node in Figure 7. The width of the resultant ranges are shown as the difference between min and max values in those figures. The ndiv parameter was set to 1,000 and 10,000 for Figures 9 and 10, respectively. The larger ndiv is, the closer the result is to the actual heater operation (800 W for 0-180 s). As we calculated with a float resolution for the temperature data, we can successfully increase ndiv up to the desired resolution to narrow down the min-max difference.

5. EFFECT OF MEASUREMENT RESOLUTION

We may not obtain such a good result as in Figure 10 for usual resolution data. Practically, a good resolution of temperature measurement may be 0.01 °C. If we throw away digits smaller than 0.01 °C from the simulated data in Figure 8, the backward simulation will stop because of a model mismatch for ndiv larger than 108, which corresponds nearly to 0.01 °C resolution as the maximum temperature change is 1.2 °C (Figure 8). Figure 11 shows the backward result for ndiv 108.

If we intend to obtain better results by increasing ndiv, we have to improve the resolution of the backward temperature data. Figure 12 illustrates our method to improve the resolution, where the value div1 is original resolution and value div2 is half of div1. Circular points indicate original A/D truncated values for the resolution of div1. If the resolution is

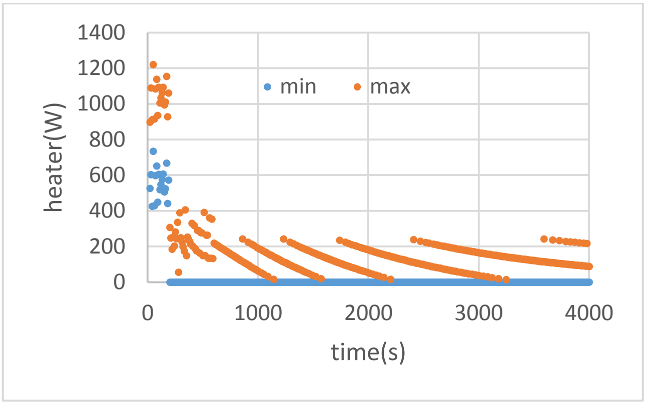

Figure 9. Backward model simulation result for ndiv=1,000.

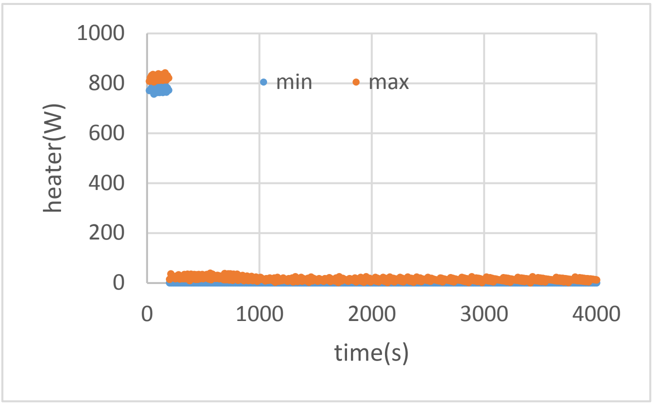

Figure 10. Backward model simulation result for ndiv=10,000.

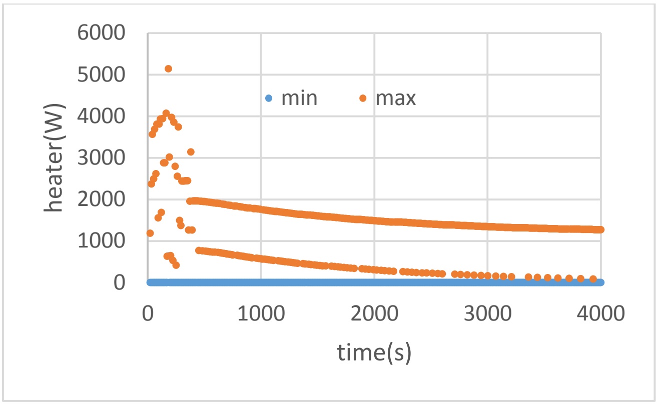

Figure 11. Backward simulation result for ndiv=108 for the resolution limited data.

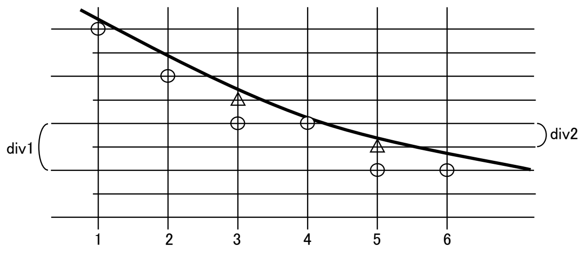

Figure 12. Resolution increase needs to raise values at some points.

improved to half of the original resolution, it is natural to raise the values at the times of 3 and 5 (indicated by triangle points). So, we wrote a shell script to raise the value when the backward simulation is stopped by detecting a model mismatch.

As shown in Figure 13, if the backward simulation detects a negative value and stops when calculating (3) or (4), we raise the value by div2 at the corresponding time t of v(t) in the case of (3) and u(t) in the case of (4). The process is repeated to get the whole sequence of backward input data pass the model match test, which means until we get a valid backward result. To improve the resolution further, the repetition is to be done for a new resolution value. There may be a case in which the model match test was passed even if the data modification was not fully done as in Figure 12. In such cases, further repetition for resolution improvement needs to add more than one resolution value and requires a longer processing time afterward.

Figure 14 shows the result for ndiv 10,000 after repeating resolution improvement. Value ranges other than the duration of heating converge to zero. Although values for the duration of heating do not converge to the correct value of 800 W, they clearly indicate heater activity and the average for the duration of 0 s to 180 s is 780 W. It shows that our method of data modification is to suppress the negative value points in (3) and (4), and not to have a resolution improvement effect at positive value points. This means that we will modify data when we detect steeper temperature falls, while we infer heating when we detect steeper temperature rises.

6. REAL DATA AND BACKWARD SIMULATION

We show the result of the backward simulator applied to the measured data in Figure 8. The backward simulator could only output a model conforming result up to ndiv=50 (Figure 15). It shows a large gap between possible minimum and maximum values. Still, the real heating power values reside within the min-max pairs. As shown in the previous section, the min value will go up and the max value will go down when the resolution of temperature data improves.

The repetitive processing of temperature data modification described in the above section improves the result, shown in Figure 16 for ndiv 10,000. We cannot expect further improvement even if we set ndiv larger, because the points of min and max are very close to each other at almost all sample times.

Checking the detail of the measured data in Figure 8, we found that there are fluctuations like the sequence of circle points in Figure 17, which would be the result of sampling and truncating quantization of the solid line. Those may be LSB (least significant bit) fluctuations caused by noises around digital thresholds. Such setbacks are treated as the results of

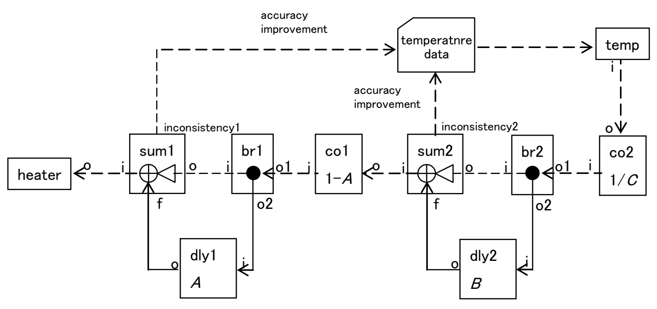

Figure 13. Backward simulation with resolution improvement.

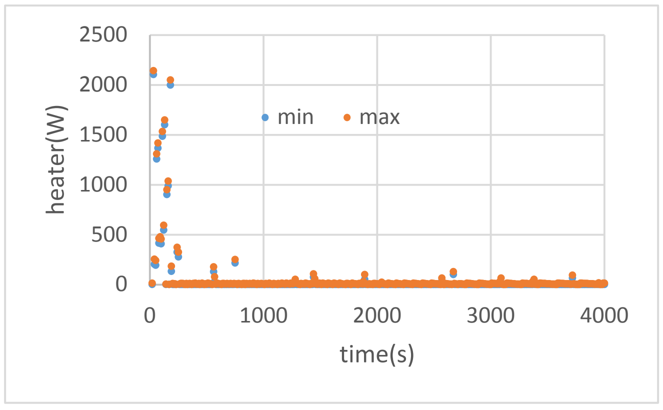

Figure 14. Backward simulation result for ndiv=10,000 with applying resolution improvement.

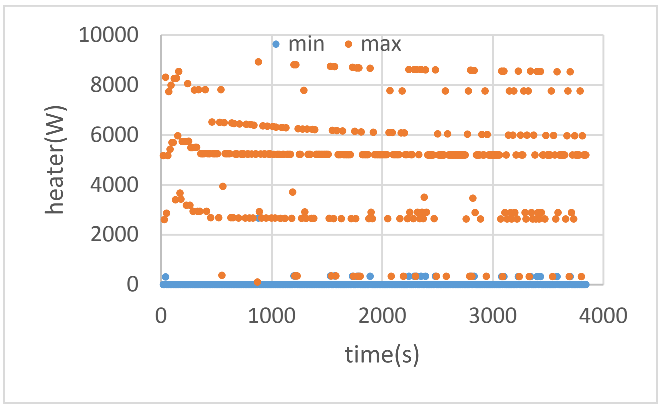

Figure 15. Backward simulation result for real data from Figure 17 for ndiv=50.

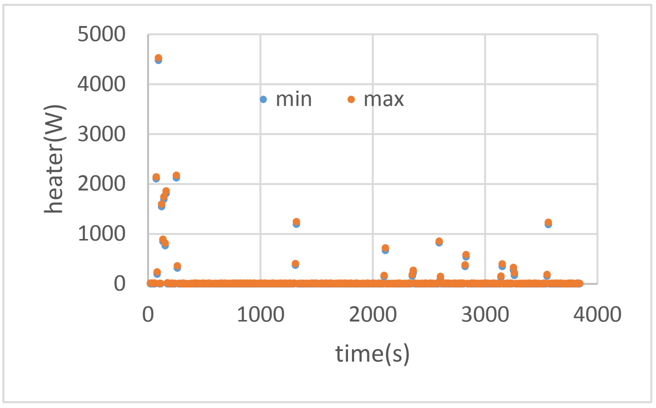

Figure 16. Backward simulation result for real data from Figure 17 for ndiv=10,000 with resolution improvement.

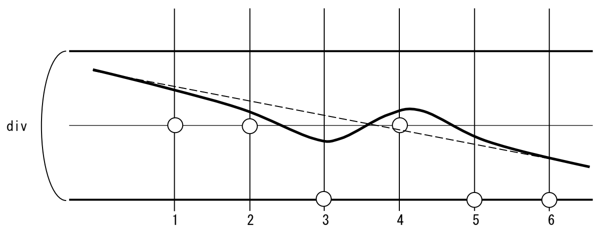

Figure 17. Real A/D data of Figure 17 have fluctuating sample points.

some heating and the simulator making spontaneous wattage rise in the resultant heating estimation in Figure 16.

If we suppress those fluctuations, we can eliminate spontaneous values. It is natural to estimate the original temperature change as the dashed line in Figure 17. Then, on probation, we eliminated them, judging by eye with adding only one LSB or subtracting only one LSB around fluctuation points, leaving the points where the fluctuation is more than one LSB. The result is shown in Figure 18, slightly smoother than the measured data in Figure 8, especially in the tail part..

Figure 19 is the result of the backward simulation of the data in Figure 18 for ndiv=10,000. It shows a better result than in Figure 16. Values are large for the period from 70 to 160 seconds and almost zero after 170 seconds. The average heater power for the duration of 0 s to 180 s is 746 W. Further improvement with larger ndiv cannot be expected because the min and max points almost coincide for all sample times.

7. DISCUSSION

By using the hybrid simulation model in a feedback type simulation, no case branching is required in the backward simulation. The backward simulator we use is for functional evaluation and includes GUI for monitoring and manual operations. It performs a single simulation for about 3.0 s for 100 sample data and about 3.6 s for 382 sample data. The processing time is not proportional to the number of samples.

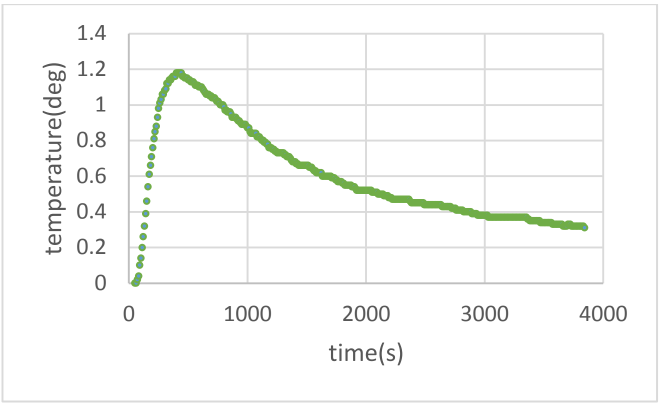

Figure 18. Fluctuation eliminated temperture data correspoinding to the measure data in Figure 9.

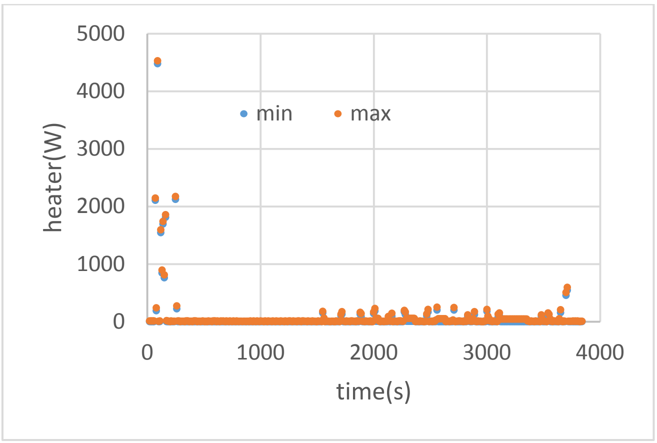

Figure 19. Backward simulation result of Fig.18 for ndiv=10,000 with resolution improvement.

One cause for it may be the fact that we used multithread Java processing on a four-core eight-thread CPU (Intel i7-3770).

Figure 20 shows the processing time for the total repetition of model mismatch and data modification relative to ndiv. Processing time in a case of model mismatch depends on the time when the mismatch is detected. Average processing time for one backward simulation is from 2.5 to 3 s. We modified the temperature data step by step, which means that we improved the resolution for ndiv 100, and then improved it for 200 by using the result of ndiv=100, as an example. The coordinate of Figure 20 shows the total processing time up to the abscissa ndiv values. Roughly, the logarithm of the processing time is proportional to the logarithm of ndiv.

Resolution improvement using model mismatch was successful. Although there are many possible methods for such data modification, our method to improve by half of the former resolution when a model mismatch is detected was shown effective. Indeed, the curve of Figure 18 is in agreement with the measured data in Figure 8. Our resolution improvement can be considered as a method to search feasible temperature data for a higher resolution.

As the estimated parameters *A* and *B* may have some errors, they may cause effects on the results of the backward simulation. We have to evaluate such effect, though they may have little effect on the long range of time sequence because those parameters were determined by a relatively long part of the temperature data.

Although we show here only one real data simulation result, the simulation model is simple, and the result for simulated data is perfect. Also, the result for real measured data is almost perfect even if our model and estimated parameters are not perfect. So, we expect similar result would be obtained for other measured data.

To improve estimations at periods of heating, we need another simulation model, e.g. a model which restricts heater wattage to one of two ranges. In such case, we need to find the time point where the estimated heating is not feasible and to determine which temperature data should be modified.

8. CONCLUSION

We showed that a backward simulation incorporating model conformation and resolution improvement is very effective. It is also shown that the backward simulation can be efficiently performed by using our hybrid method (forward simulation objects in backward simulation structure) by eliminating case

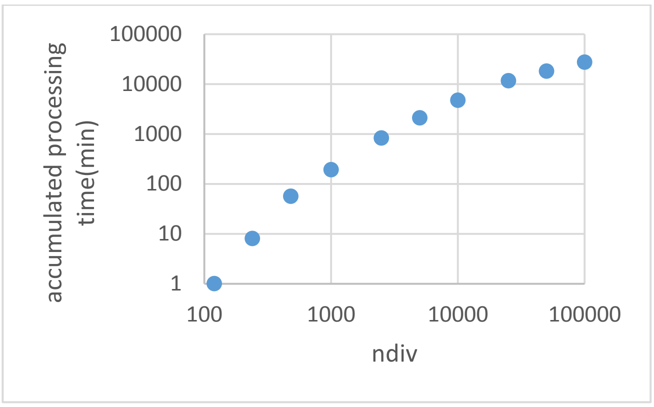

Figure 20. Processing time including resolution improvement vs ndiv for the cases in Section 5.

branches. Causal input changes can be practically determined for the simulated data and for the real measurement data. The internal state of any system can be determined by the backward simulation if we define the system's model properly for backward simulation.

In the case of data with high resolution, the backward simulator outputs almost perfect results. For cases of limited resolution, repetitive resolution improvement responding to model mismatches was successfully done. We also showed that the backward simulation with real measurement data can be effectively done, although noise elimination was needed for errors larger than quantising errors.

The results suggest that we can separate noise from signal by using the backward simulation with a model conformation test. The backward simulation offers us a new method to infer causal inputs and internal states of various systems. We need to study the relation among the precision of simulation output, degree of model fitness and SNR of data for backward processing.

ACKNOWLEDGEMENT

This work was supported by JSPS KAKENHI Grant Number 25540006. We thank one of the reviewers who motivated us to improve the simulation.

REFERENCES

[1] J. V. Beck, B. Blackwell and C.R.S. Clair Jr., Inverse Heat Conduction Ill-posed Problems, John Wiley & Sons, 1985, ISBN 0-471-08319-4.

[2] Y. Jarny and D. Maillet, Linear Inverse Heat Conduction Problem – Two Basic Examples, http://www.sft.asso.fr/Local/sft/dir/user-3775/documents/actes/Metti5_School/Lectures%26Tutorials-Texts/Text-L10-Jarny.pdf.

[3] K. Oguni, Inverse problem and instrumentation, Ohmsha, Tokyo, 2011, ISBN 978-4-274-06829-4.

[4] T.B.Bako and T.Daboczi, Improved-Speed Parameter Tuning of Deconvolution Algorithm, IEEE Trans. Instrum. Meas., vol.65, no.1, pp.1568-1576, 2016.

[5] A.Muqaibal, A,Safaai-Jazi, B.Woerner and S.Raid, UWB Channel Impulse Response Characterization Using Deconvolution Techniques, Proc. MWSCAS, 2002.

[6] Chueng-Chiu Huang and His-Huang Wang, Backward Simulation with Multiple Objectives Control, Proc. IMECS (International MultiConference of Engineers and Computer Scientist), Hong Kong, 2009.

[7] C.D.Twigg and D.L.James, Back Steps in Rigid Body Simulation, ACM trans. Graph., vol. 27, no.3, article 25, 2008.

[8] J.J.Cook, Reverse Execution of Java Bytecode, Computer Journal, vol.45, no.6, pp.608-619, 2002.

[9] Y. Hiranaka and T. Taketa, DESIGNING, BACKWARD RANGE SIMULATOR FOR SYSTEM DIAGNOSES, Proc. XX IMEKO World Congress, 2012.

[10] Y. Hiranaka., H. Sakaki, K. Ito, T. Taketa and S. Miura, Numerical Backward Simulation Model with Case Branching Capability, Proc. 4th International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH 2014), pp.225-230, 2014.