



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, 992-8510 Yonezawa,, Japan

ABSTRACT

We are developing a backward simulator, which determines unknown system input from system output by using precise system model. However, its processing time would increase unlimitedly if the simulation model requires multiple case branching, which is natural to 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 a 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: Acta IMEKO, vol. A, no. B, article N, month year, identifier: IMEKO-ACTA-0A (year)-0B-0N

Section Editor: name, affiliation

Received month day, year; **In final form** month day, year; **Published** month year

Copyright: © year IMEKO. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited

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

Inverse problem which estimates input values or internal states from output values can be converted to simultaneous linear equations if it is linear problem. Generally, we cannot solve them with enough accuracy because output values have measurement errors which include quantization errors [1]. Generally, to bring in restrictions on intermediate values, e.g. non negative, and causality that current value do not affect past values may improve those estimation. As a way to bring in such condition, we are studying backward simulation.

Backward simulation is to start with the outputs of the system and to trace back by applying system model. By executing a simulation, it would make clear the conditions of intermediate values and easy to find possible inputs or internal state, efficiently. Therefore, it is used in many fields such as process scheduling [2], initial position estimation of physical objects [3], and software debugging [4].

Though it is theoretically applicable to the field of indirect measurements, it may have some difficulties in creating backward system model. Furthermore, a typical backward simulation requires case branching when it has multiple possibility of conditions on its backward trace. However, we have experienced to cope with them [5,6], and showed that some physical restriction may effectively reduce the number of case branches.

In this paper, we show a method to estimate the heating time and heating power by executing backward simulation by applying time sequence of room temperature. Although thermal conduction is a linear problem which can be solved by applying inverse matrix, small singular values of its system matrix make the problem as an ill-posed one which make the results sensitive to measurement errors and unstable [7, 8]. Our method processes backward values always as ranges of (the minimum and the maximum) values, which indicate error range. Also, it can execute an integration loop as a no branching process in the backward simulation.

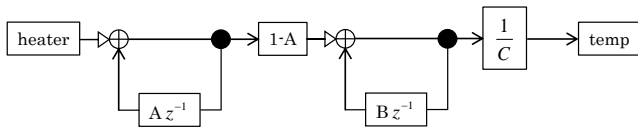


Figure 1. Forward simulation model of room temperature.

Hereinafter, we describe the concept of backward simulation, hybrid implementation of backward integration loop, simulation results for error free temperature data, effect of quantization error and countermeasure to it, and simulation results for real measurement data.

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. Blocks 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 after the time of turning off the heater. The first integration loop corresponds to the neighbor of the heater that accumulates calorie 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 calorie to temperature. The response can be expressed as $T=a(e^{-bt} - e^{-ct})$, where a, b, c are constants, and t represents time.

Figure 2 shows the implemented software objects and connection diagram for the simulation of Figure 1. Each object has corresponding function shown in Figure 1, “heater” for heating for specified time duration, “wsum” for summing calorie with wattage input port, “csum” for summing calorie, “br” for branching with the same output value, “dly” for one sampling time delay, “co” for coefficient multiplier, and “temp” for temperature recording. These objects are coded in Scala classes, which use 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 values of each object.

Data flow on the connection links consist of time and values. As an example, heating power 800W from “heater” at its output port “o” at the time of 10.0s is shown in a XML style UCF (universal communication format) message [5,6] `<sim><s>heater<s>o</s></s><t>10.0</t><v>800.0</v></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 [6]. The `<s>` tag indicates the source of the message. The source tag is nested in this case to mean “o” of “heater.”

In the backward simulation (Figure 3), UCF messages flow in the backward direction indicated by dashed lines with

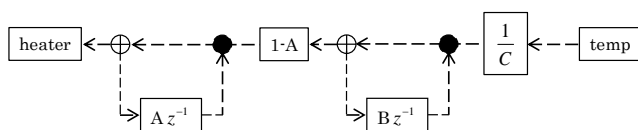


Figure 2. Backward simulation model corresponding to Figure 1.

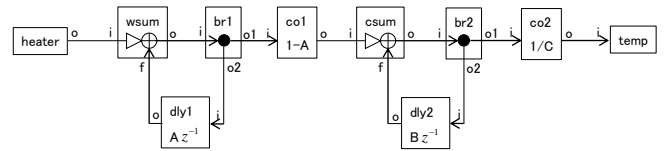


Figure 3. Forward simulation objects and connections corresponding to Figure 1.

reversed arrows. The node “temp” is the starting block which sends the time sequenced temperature in the reversed order, packing time and temperature 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 [6].

By narrowing the range, we can control the resolution of 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` equals 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 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 $y(i)=y(i-1)+x(i)$.

Fully synchronized simulator will do the work naturally. However, we intend to perform our simulation in a distributed processing environment, and we preferred asynchronous processing as long as it is possible. Then, the summation node is designed to have input record table which keeps time and value pairs received at the left input port (triangle arrowed port in Figure 1-15). The data message from the feedback port (from the delay node) triggers summation process by picking up the data of the same time from the input record table.

In Figure 5 which shows the backward version of Figure 4, we need to calculate two backward outputs from a single backward input at the summing point, satisfying $y(i)=x(i)+f(i)$. The computing intensive solution is to simulate all the pairs of $x(i)$ and $f(i)$, matching the equation. A practical solution is to perform simulation for finite number of divided range pairs, e.g. $\{x, f\}$ for $\{0.0$ 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 [6]. Case branching mechanism is needed for such processing.

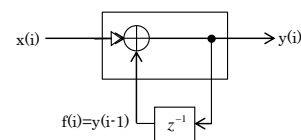


Figure 4. Forward integration loop.

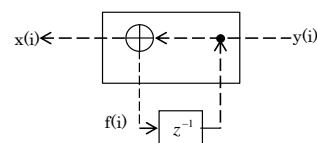


Figure 5. Backward integration loop.

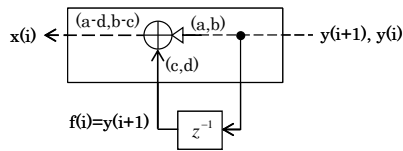


Figure 6. Hybrid backward integration loop.

However, if we apply another solution to use forward simulation object in the backward simulation as in Figure 6, we can avoid the expansion of processing time caused by the case branching. In Figure 6, we can formulate the process as $x(i)=y(i)-f(i)$ and $f(i)=y(i+1)$. The same input record table used in Figure 4 can be utilized for making $y(i)$ available to the reverse delayed feedback triggered by $f(i)=y(i+1)$.

In Figure 6, the dot node sends received backward data in time reverse order through the two links to the sum node and to the delay node. And the delay $y(i+1)$ will be treated as the feedback $f(i)$. 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 treat 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 calorie value. 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 case.

Figure 7 shows the resulting practical hybrid backward simulator, and Figure 8 shows implemented objects and connections. Simulation objects in Figure 8 are the same objects as in Figure 2, but 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 8 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 not matching time data coming from the other backward port. Still, the simulator we made has forward and backward simulation functions.

We describe here the mechanism of model mismatch detection in detail. Sum modules (wsum and csum) in Figure 8 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 may be expressed as

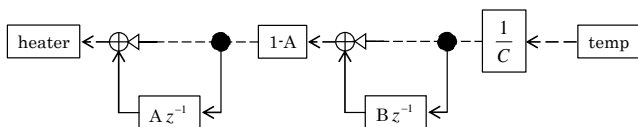


Figure 7. Hybrid backward simulation model.

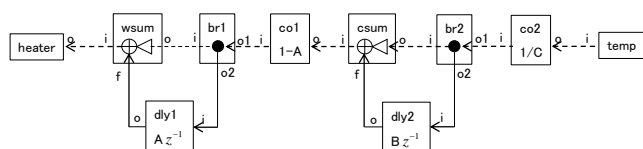


Figure 8. Hybrid backward simulation objects and ports.

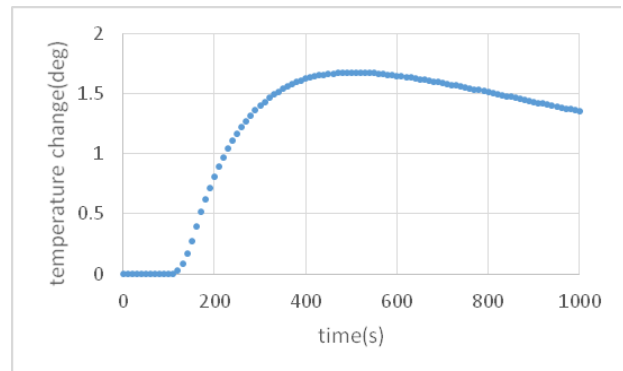


Figure 9. A simulated temperature change caused by heating (100-150s).

$Bv(t-1)$ because dly2 node delays the backward flow signal for one sample time and multiplied by constant coefficient B . The backward output $u(t)$ from csum is expressed in equation (1) and must be positive or zero for it expresses delivered calorie.

$$u(t) = v(t) - Bv(t-1) \geq 0. \quad (1)$$

The backward output from wsum can be expressed in equation (2),

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

If the both inequality 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 with the models of the simulation.

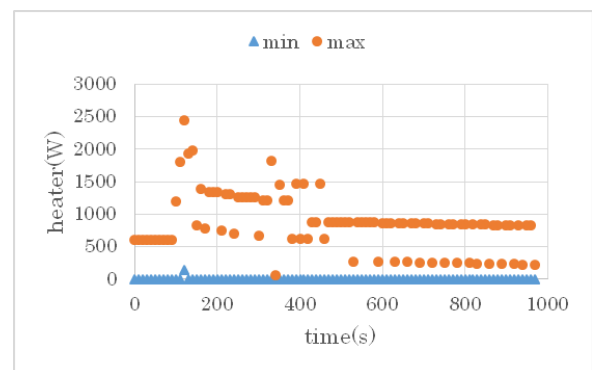


Figure 10. Backward model simulation result for ndiv=100.

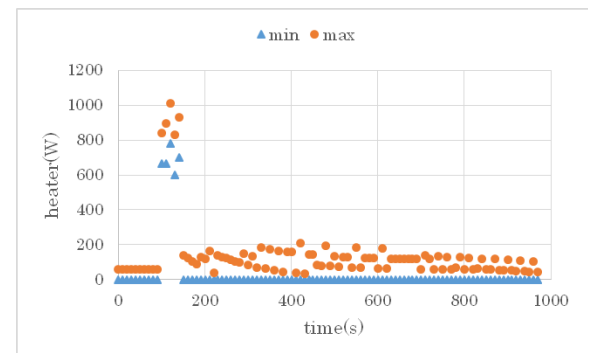


Figure 11. Backward model simulation result for ndiv=1,000.

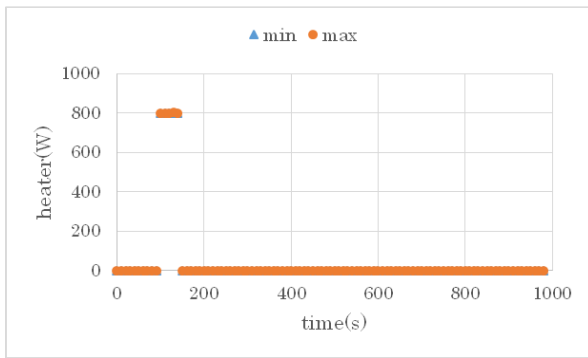


Figure 12. Backward model simulation result for ndiv=100,000.

4. MODEL SIMULATION

At first, we verified the correctness of the simulator. Figure 9 is a result of room temperature calculated by forward simulation where the heater (800W) was turned on for the duration from 100s to 150s. The parameters $A(=0.93)$, $B(=0.994)$ and $C(=4500 \text{ cal/deg})$ were estimated by the real measurement shown in the next section.

Figures 10-12 are the results of backward simulation when the temperature change in Figure 9 was fed backward from “temp” node in Figure 8. The width of resultant ranges are shown by the difference between min and max values in those figures. The ndiv parameter was set 100, 1,000 and 100,000 for Figures 10, 11 and 12, respectively. Although the smaller ndiv makes coarse inference, larger ndiv estimates close to the original situation (Figure 12). As we used float type calculation for the temperature data, we can successfully increase ndiv up to practical desired resolution to narrow down min-max difference.

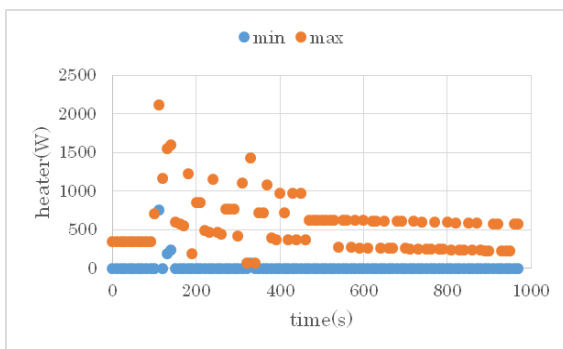


Figure 13. Backward simulation result for ndiv=170 for the resolution limited data.

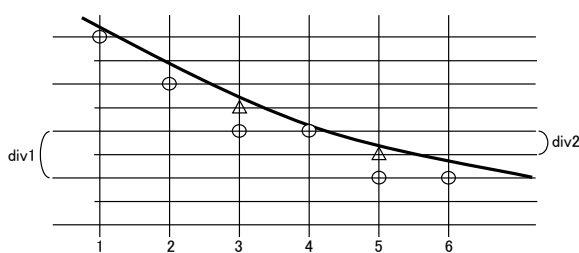


Figure 14. Resolution increase needs to raise values at some points.

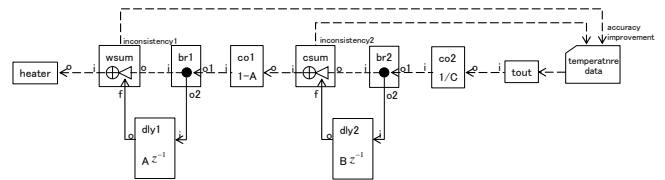


Figure 15. Backward simulation with resolution improving processes.

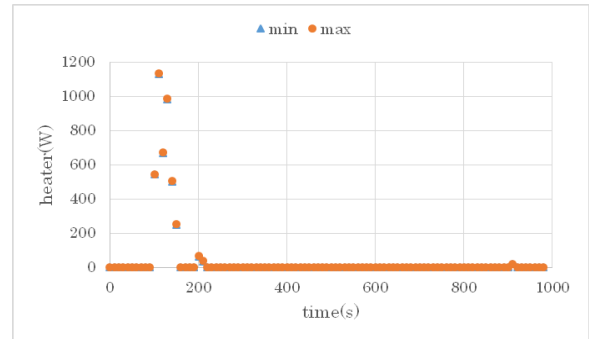


Figure 16. Backward simulation result for ndiv=50,000 after resolution improvement.

5. EFFECT OF RESOLUTION OF MEASUREMENT

We may not obtain such a good result as in Figure 12 for usual resolution data. Better resolution of temperature measurement may be $0.01 \text{ }^\circ\text{C}$. If we throw away digits smaller than $0.01 \text{ }^\circ\text{C}$ from the data in Figure 9, the backward simulation will stop because of model mismatch for ndiv larger than 170, which corresponds $0.01 \text{ }^\circ\text{C}$ resolution as the maximum temperature change is 1.7°C (Figure 9). Figure 13 shows the backward result for ndiv 170.

If we intend to obtain better results for larger ndiv cases, we have to increase the resolution of the backward temperature data. Figure 14 illustrates our method to increase the resolution, where the value div1 is original resolution and value div2 is a half of div1. Circular points indicate original A/D truncated values for the resolution of div1. If the resolution is upgraded to half, it is natural to raise values at the time of 3 and 5 (indicated by triangle points). So, we wrote a script to raise the value when the backward simulation is stopped by detecting model mismatch.

In detail as shown in Figure 15, if the backward simulation detects negative value and stops when calculating equation (1) or (2), we raise the value by one div2 at the corresponding time t of $v(t)$ in the case of equation (1) and $u(t)$ in the case of equation (2). The process is repeated to get the whole sequence

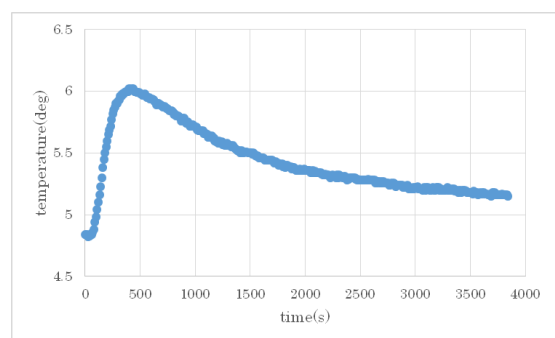


Figure 17. A measured temperature change caused by a real heater (ON for 0-180).

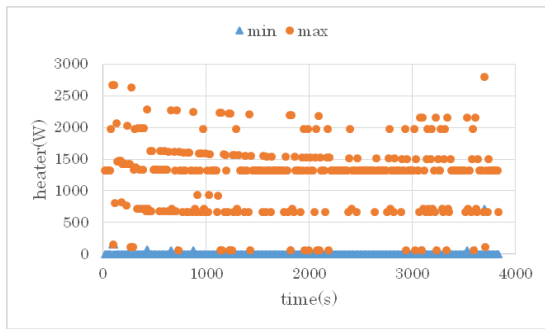


Figure 18. Backward simulation result for real data of Figure 17 for $ndiv=50$.

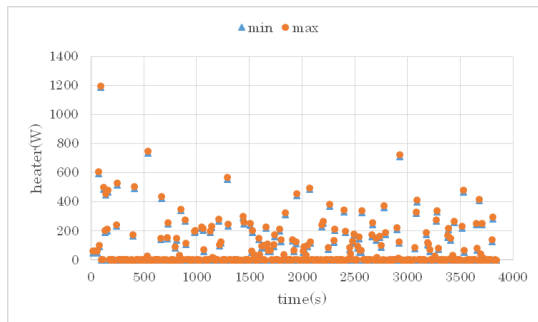


Figure 19. Backward simulation result for real data of Figure 17 for $ndiv=10,000$ with resolution improvement.

of backward input data pass the model match test, which means that we got a valid backward result. To increase the resolution further, the repetition was redone for a new resolution value. There may be a case in which model match test was passed even if the data modification was not fully done as in Figure 14. In such cases, further repetition for resolution improvement needs to add more than one resolution value and requires longer processing time.

Figure 16 shows the result for $ndiv=50,000$ after repeating resolution improvement. Value ranges at other than the duration of heating converge to zero. Although values for the duration of heating do not converge to the correct value of 800W, they clearly indicate heater activity. It shows that our method of data modification is to suppress negative value point in equations (1) and (2), and is not to make resolution improvement effect at positive value points.

6. REAL DATA AND BACKWARD SIMULATION

We applied the backward simulator to real data in Figure 17, which is room temperature response to 800W infrared heater turned on for 180 seconds in a tiny room of 3.6 m³. The sensor is Sensirion's SHT71, which has 0.01 degree resolution, placed near the sensor at 1m high in the room.

The backward simulator can only output a model conforming result up to $ndiv=50$ (Figure 18). It shows large area of possibility between the min-max values, although it keeps real heating values within its min-max pairs.

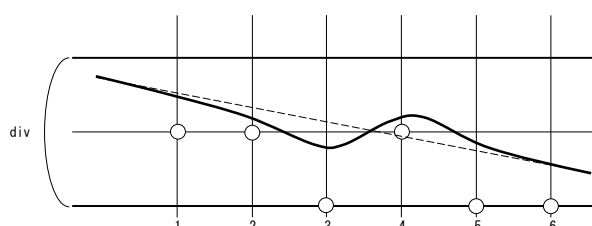


Figure 20. Real A/D data of Figure 17 have fluctuating sample points.

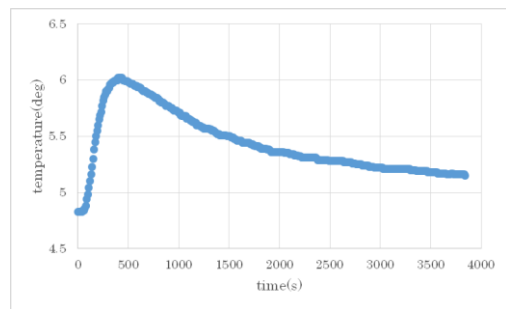


Figure 21. Fluctuation eliminated temperature data corresponding to Figure 17.

The repetitive processing of temperature data modification described in the above section improves the result, shown in Figure 19 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 Figure 17, we found that there are fluctuations like the sequence of circle points in Figure 20 which would be results 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 must be treated as results of some heating and temperature simulator make the spontaneous wattage raise in the resultant heating estimation appeared in Figure 19.

We cannot correctly eliminate such fluctuations as we don't know exact noise values. However, any suppression of fluctuation would eliminate spontaneous values. Then, on probation, we eliminate them, judging by an eye with adding one LSB or subtracting one LSB around fluctuation points by estimating original temperature as the dashed line in Figure 20. One result is shown in Figure 21, slightly smoother than Figure 17 without setbacks.

Figure 22 is the result of backward simulation of Figure 21 for $ndiv=10,000$. It shows better result than in Figure 19. Value ranges are near to zero after 300 seconds and high for the period from 50 to 230 seconds. Further improvement with larger $ndiv$ cannot be expected because the points min and max are almost coincide for all sample times.

7. DISCUSSION

By using the hybrid simulation model in a feedback type simulation, no case branching is required. The backward simulator we used is for functional evaluation and includes GUI for monitoring and manual operations. It performs single simulation for about 3.0 sec for 100 sample data and about 3.6

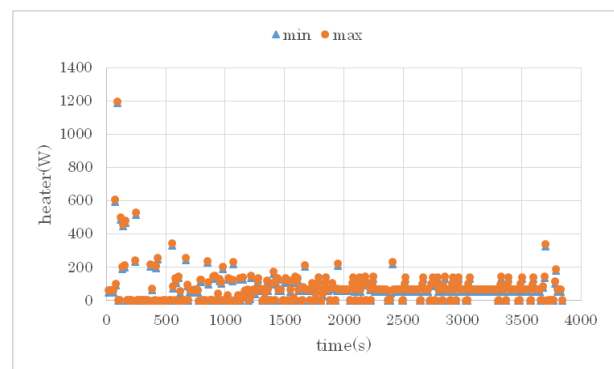


Figure 22. Backward simulation result of Fig.21 for $ndiv=10,000$ with resolution improvement.

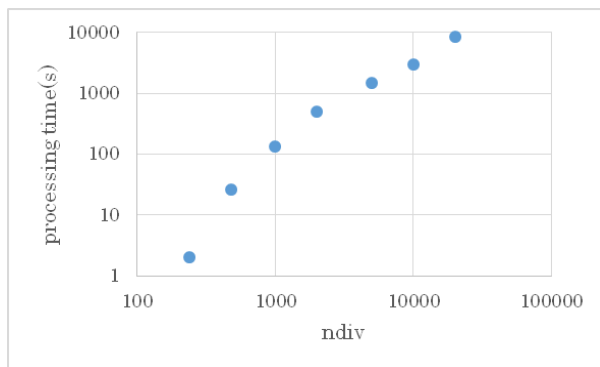


Figure 23. Processing time including resolution improvement vs ndiv for the cases in the section 5.

sec for 382 sample data. Processing time is not proportional to the number of samples. One cause for it may be the fact that we used multithread processing on a four-core eight-thread CPU (Intel i7-3770).

Figure 23 shows 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 mismatch detected. Average processing time for one backward simulation is from 2.5 to 3 seconds. We modified the temperature data step by step, which means that we improved resolution for ndiv 240, and then improved it for 480 as an example. The coordinate of Fig.23 shows total processing time up to the abscissa ndiv values. Roughly, the logarithm of processing time is proportional to the logarithm of ndiv for larger ndiv values.

Resolution improvement using model mismatch was successful. Although there are many possibility for such data modification, our method to increase half the former resolution when model mismatch detected was shown effective. Actually, the curve of Figure 21 do not have an excessive difference from Figure 18. 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 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.

To improve estimations at periods of heating, we need another simulation model, e.g. one which restricts heater wattage to be one of limited number of values. In such case, we need to find the time point where the estimated heating is out of possibility and to determine which temperature data should be modified.

8. CONCLUSION

We showed that backward simulation can be efficiently performed by using hybrid method (forward simulation objects in backward simulation structure), eliminating case branches. Causal input changes can be practically determined for the simulated data and for the real measurement data. Internal state of any system can be determined by 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 result. For cases of limited resolution, repetitive resolution improvement responding to model mismatches was successfully done. And also, we showed that the backward simulation with real measurement data can be effectively done, although noise elimination was needed.

The results suggest that we can separate noise from signal by using backward simulation with 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 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.

REFERENCES

- [1] K. Oguni, Inverse problem and instrumentation, Ohmsha, Tokyo, 2011, ISBN 978-4-274-06829-4.
- [2] Chueng-Chiu Huang and His-Huang Wang, Backward Simulation with Multiple Objectives Control, Proc. IMECS (International MultiConference Engineers of Engineers and Computer Scientist), Hong Kong, 2009.
- [3] C.D.Twigg and D.L.James, Back Steps in Rigid Body Simulation, ACM trans. Graph., vol. 27, no.3, article 25, 2008.
- [4] J.J.Cook, Reverse Execution of Java Bytecode, Computer Journal, vol.45, no.6, pp.608-619, 2002.
- [5] Y. Hiranaka and T. Taketa, DESIGNING, BACKWARD RANGE SIMULATOR FOR SYSTEM DIAGNOSES, Proc. XX IMEKO World Congress, 2012.
- [6] 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.
- [7] 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.
- [8] 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.