XX IMEKO World Congress
Metrology for Green Growth
September 9–14, 2012, Busan, Republic of Korea

DESIGNING BACKWARD RANGE SIMULATOR FOR SYSTEM DIAGNOSES

Yukio Hiranaka and Toshihiro Taketa
Yamagata University, Yonezawa, Japan, zioi@yz.yamagata-u.ac.jp

Abstract: We propose a simulator for system diagnoses, which traces backward from output to input for finding undesirable situations. By designing such a simulator with XML type data format called UCF, we can make it as a forward and backward capable simulator. We describe the principle, implementation, and an example of its application to electric power saving mechanism which uses dynamic pricing.

Keywords: backward simulator, XML, UCF, smart grid.

1. INTRODUCTION

As systems are getting larger and more complex, it is harder to find problematic behaviours, which may be rare in an ordinary operation. It would be also hard to come across them by any simulation without appropriate insight to the system. However, we can easily state undesirable outputs of the system. Then, it is good to start from such outputs, trace back to the inputs and check the situation whether it can occur or not. Huang and Wang shows a backward and then forward simulation for constrained scheduling[1]. Backward simulator can be used as a simulator starting with a marginal output to trace back to input conditions. We propose a backward simulator which can handle practical analog trace back by using range signals.

2. EXAMPLE OF SYSTEM UNDER TEST

We introduce a dynamic pricing mechanism of smart grid[2] as an example to be tested by our simulator. Figure 1 shows the control loop where the price controller raises the price of electric power dynamically to suppress the total usage within the capacity. However, the power distributor would lower the price to increase the usage near to the capacity in order to maximize their revenue. Zhong[3] and Chen et al.[4] show that the scheme is stable by setting appropriate price function and user response function with continuous model simulation. Mohsenian-Rad et al.[5] states that his simulation is valid for asynchronous demand responses when power consumers do not response simultaneously to the control from the power distributor. However, we have to test all the possible situations including synchronous demand response cases. Instinctively, we should be cautious about the oscillation in which many users stop usage after the price raised and restart usage when the price lowered after the controller detects significant decrease of usage[6].

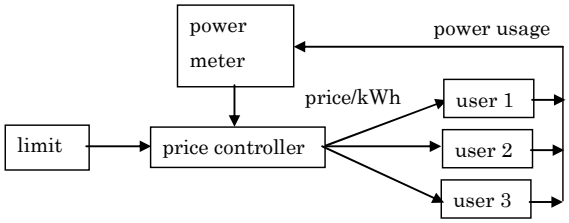

Figure 1. Electric power system with dynamic price control.

To verify such phenomenon by a simulator, we may need to repeat simulations with vast set of conditions, in which the phenomenon might occur by chance. It is more reliable to start with the failure status such as over usage of power and to trace back the price range which can cause the over usage.

3. BACKWARD SIMULATOR

Fig. 2 shows our problem to solve. It is easy to identify two input set of A and C, which will be always processed to be good results or not good results. The input set B is our main concern which may be processed into not good results even we do not recognize the existence of such cases. The backward simulation is expected to start from a not good result to catch possible input set for it.

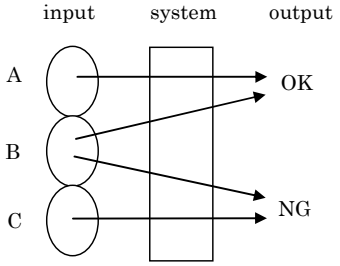

Figure 2. Some range of input may cause system failure.

To enable such backward simulation, we have to define backward process corresponding to the forward process, formalized in Fig.3. There may be easy systems and not so easy systems for defining backward processes.

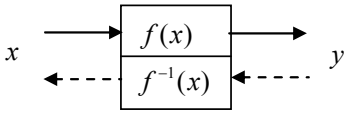

Figure 3. Forward and backward processes.

Fig.4 is an example of not so easy system component which just adds two inputs. We cannot determine each input value from the output. However, we can determine the range for each input ($0 < x_1, x_2 > y_u$) which are consistent with the output, because any input value is confined within a certain physical signal range. In this case, $x_1 > 0$ and $x_2 > 0$.

We do not use the relation between the input signals in this paper to simplify the simulation. Then, our simulation outcome may be getting broader. On the other hand, there is narrowing components such as a fan out component. In Fig.5, three different backward range signals meet at the component. The inputs must be in the overlapping range of three range signals as $\max(x_1, x_3, x_5) < x < \min(x_2, x_4, x_6)$.

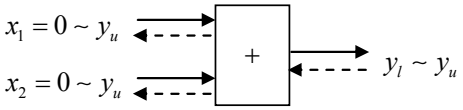

Figure 4. Output range to input range constraint.

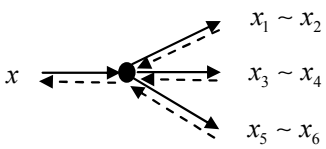

Figure 5. Backward range curtailment.

There are two terminating points for the backward simulation. One is the input port of the system. If the resulting backward range signal at an input port is outside of the possible input range, it means the simulation terminated with the negative conclusion which tells us that the starting output does not occur on the system. The other terminating point is the feedback point to an already determined component port such as in Fig.6. If the feedback range signal is outside of the previously determined range, it means the negative conclusion. If we do not meet any disparity even after all the backward simulation, it means that the starting output may occur. Of course, we must consider the dynamic behaviour of the system and we might have to repeat the backward feed back simulation to test whether we meet range disparity or not.

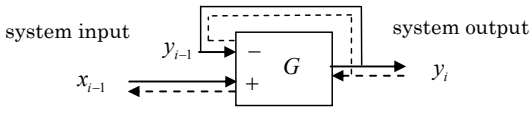

Figure 6. Backward range simulation with feedback loop.

To make such a back trace, we need a backward flow of certain information through the simulated components. Figure 7 shows the user component of Fig.1 which is comprised with forward process (fproc), backward process (bproc), and an internal state such as the current power usage.

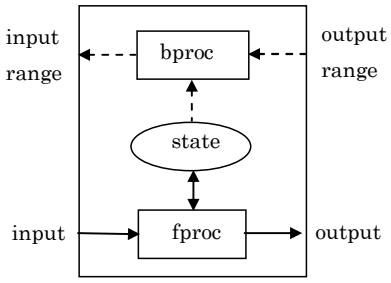

Figure 7. Inner components of user component.

Forward process is a mapping or conversion from input to output. However, in the backward simulation, input value may not be mapped or converted to a single value. A differential component need initial value to be added to the integral of the output signal. However, there will be certain lower and upper limits for inputs. Then, we can determine the input at the lowest initial value and the input at the highest initial value. So, we can design a backward simulator which deals with range (lower bound, upper bound) as the signal in the backward flow.

Also, we should design backward process of each simulated component with considering the characteristics of them. First of all, transfer functions of each component are strongly desired to be monotonic. A multi-input component need division of output range to input ranges. As we are dealing with the worst case analysis, it is natural to select the division to concentrate to a single input or uniformly distributed. To test all the cases, conditions are to be stored in the simulator and scheduled separately.

An integrator component needs to have integrated value as its internal state. A fan out component must check the backward signals whether they have overlap or not. If there exists no overlap for any output pairs, it means that the situation does not occur and the starting output of the simulation does not appear. Such possibility check can also be done at the components which have some internal state.

Backward simulation kernel is to be designed to know the connection of the components and to operate by the following principles:

(1) Send range signals backward through the connections.

(2)When backward process have choices for some inputs, trace back in each case.

(3)When the backward range signals terminated at system's input end, check whether the range signals exclude the real input or not. If any one of the inputs is outside the corresponding range signal, it means that the starting output cannot appear. Otherwise, it may appear.

(4)In a system which has loop, the range signal will come over to some predetermined output. If the resulting range after the loop calculation excludes the predetermined range, it means that the starting output cannot appear. Otherwise, we need to repeat the back trace until we are convinced that the range signal does not flip to exclude the predetermined output range.

4. IMPLEMENTATION

Figure 8 is our implementation of the user component of Fig.1, which has forward and backward simulation capability by using Actors of programming language Scala. In our backward simulation, output ports accept signals and input ports send out signal. Then, we designed an inner object dst which directs input signals and output signals to the specified destination. To enable this, signals must carry the information which indicates their destination.

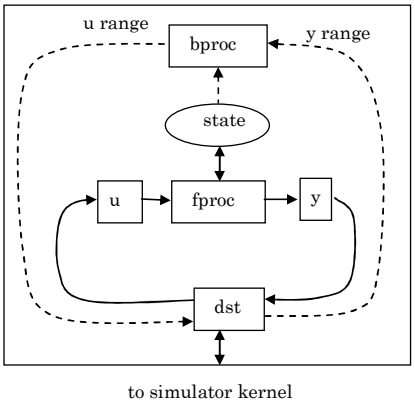

Figure 8. UCF implementation of user component.

It is natural to express the whole signal using character strings to include the destination information for the simulation. We are expressing it in UCF (Universal Communication Format [7-10]) which is an XML style data representation comprised of destination tag and content to be delivered. As an example, input 1.1 to the input port i of component A is expressed by

<A>1.1</A>.

The output from the component is expressed by

<sim><s>A<s>o</s></s>5.0</sim>,

where sim is the simulator kernel which accepts and redirects the message and proceed the simulation steps. The tag part s shows the source of the message, which can be self included to indicate the source to be o inside of the component A. The self included source expression is a convenient way to process by structured components.

Backward message to inquire the input range corresponding to output range "1.0< o <5.0" is expressed by

<A><o>1.0,5.0</o></A>,

where o denotes the output port. The response to this inquiry can be expressed by

<sim><s>A<s>i</s></s>4.0,Infinty</sim>,

where i denotes the input port and the message tells that the widest possible input range is "4.0< i < ∞ ".

5. SIMULATION EXAMPLE

Figure 9 shows a simplified simulation for the case of Fig. 1 with a single aggregated user component. Single user configuration is equivalent to simulate synchronous user responding, which is the most severe condition for the price controller. Such situation would be easily appear when an optimally programmed home energy controller is deployed in every consumer site.

The pricing function is $p=ku/C$, where price p is proportional to the total power usage u with coefficient k and the total power capacity C . The user function is $y = y_{old} + \alpha(w - y_{old} \cdot p)$, where y is the power usage by the user, y_{old} is the previous value of y , α is the coefficient adjusting convergence speed, and w is the user's parameter for acceptable hourly cost. The component manin is used to inject a starting output value.

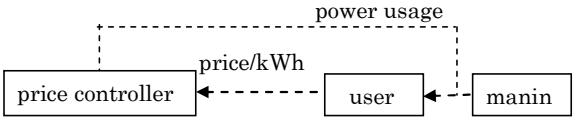

Figure 9. Simplified backward simulation.

Fig.10 shows an example of simulation results. The left part indicates the source of message. The second part indicates backward time sequence. The remaining part shows the range signal or some information from the simulation components.

The second line in Fig.10 shows the starting range of over usage of 25kW (lower and upper bounds are same value in this case). The third line shows the calculated output from user backward process, which means price input must be 0 at the lowest when y_{old} is 15kW, and 2.5 at the highest when y_{old} is 20kW. Consequently, the fourth line shows that the necessary input to the price controller (previous output of user) must be in the range of [0, 2.5]. It is looped back to the output of the user and the range does not include [15, 20] of the third line range, which means that the case cannot occur. Figure 11 is a case of attainable output, which repeats for ever.

```
I              user     w=100.0 alpha=0.1
Bmanin  100    25,25
Buser   99.0   p=0.0 at yold=15.0, p=2.5 at yold=20.0
Bprice  98.0   0.0,2.5
Buser   97.0   input 0.0, 2.5 is outside yold range 15.0, 20.0
```

Figure 10.. Backward simulation starting from unattainable output (OK case).

```
I              user     w=400.0 alpha=0.1
Bmanin  100    25,25
Buser   99.0   p=17.5 at yold=20.0, p=Infinity at yold=0.0
Bprice  98.0   17.5,Infinity
Buser   97.0   p=0.0 at yold=20.0, p=Infinity at yold=0.0
Bprice  96.0   0.0,Infinity
Buser   95.0   p=0.0 at yold=20.0, p=Infinity at yold=0.0
Bprice  94.0   0.0,Infinity
```

Figure 11.. Backward simulation starting from attainable output (NG case).

Usually as in Fig.10, we can determine whether the starting value is attainable or not after tracing only once through the feedback loop. By using binary search method, we can plot the border between the attainable and unattainable area for parameters W and starting y output such as in Fig.12. In this figure α =0.1. The lower right area is unattainable, and the area y<=20 is normal area as the

consumption is under the power limit of 20kW. The upper middle area means problematic area where overload (y>20) may occur and cause the system fail. It means that overload may occur by setting hourly acceptable cost W larger than a certain amount in the user's controlling scheme.

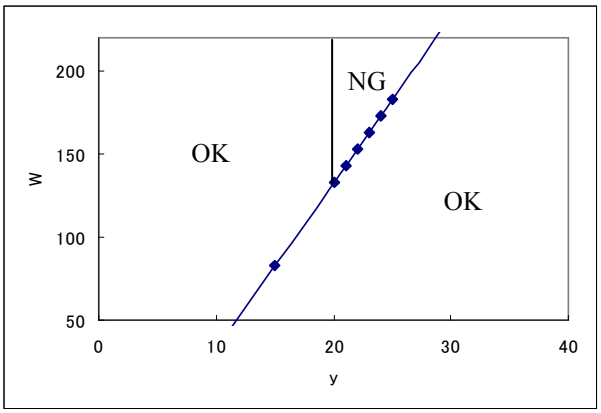

Figure 12.. Backward simulation results: the lower right and y<20 regions are acceptable..

Also, by using backward simulation, we can determine the border relationship between W and α which do not cause over load condition. Fig.13 shows the relationship in the case of starting y=21kW. This shows that we should limit hourly acceptable cost W according to convergence speed α .

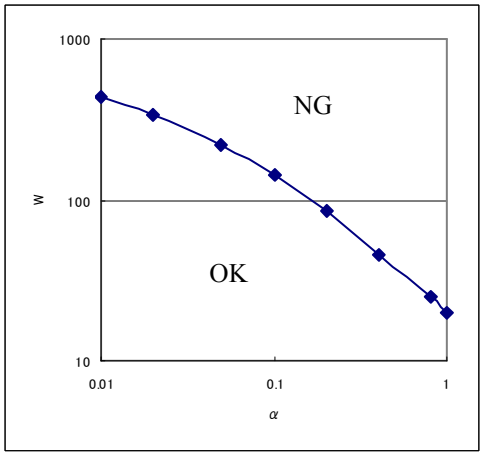

Figure 13.. The relation between max W and α ..

6. CONCLUSION

We proposed a backward simulator for diagnosing complex systems, and showed a typical example of its application to smart grid mechanism. We hope that our backward simulation and the range signal scheme are adapted to various systems, and show their power in analyzing complex systems. In the course, we have to devise mechanisms to deal with irreversible processes, appropriately.

References:

[1] 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.

[2] Iordanis Koutsopoulous and Leandros Tassiulas, Change in Demand Load Control for the Smart Grid, IEEE Network, vol., no.5, pp.16-21, 2011.

[3] Zhong Fan, Distributed Demand Response and User Adaptation in Smart Grid, Proc. Integrated Network Management, pp.726-729, 2011.

[4] Lijun Chen, Na Li,Steven H.Low and John Doyle, Two Market Models for Demand Response in Power Networks, Proc. IEEE Int'l Conf Smart Grid Comm., 2010.

[5] Amir-Hamed Mohsenian-Rad, Vincent W.S.Wong, Juri Jatskevich, Robert Schober and Albert Leon-Garcia, Autonomus Demand-Side Management Based on Game-Theoretic Energy Consumption Scheduling for the Future Smart Grid, IEEE trans. Smart Grid, vol.1, no.3, pp.320-331, 2010.

[6] Y. Hiranaka, Cross-layer Design and Control for Smart Grid, http://smartgrid.ieee.org/newsletter/december-2011.

[7] Y. Hiranaka, H. Sakakibara and T. Taketa, Universal Communication Format for Multimedia Data, Proc. Sixth International Conference on Computational Intelligence, and Multimedia Applications (ICCIMA 2005), 338-339, 2005.

[8] XML specification, www.xml.org.

[9] Y. Hiranaka and T. Taketa, Object-Oriented Framework for Cross-Layer Communication, Proc. ITSim2008 (International Symposium on Information Technology), pp.1640-1645, 2008.

[10] Y. Hiranaka, Y. Sato, T. Taketa and S. Miura, Smart Power Outlets with Cross-layer Communication, Proc. ICACT2011(International Conference on Advanced Communication Technology), pp.1388-1393, 2011.