Proceedings of the 1999 IEEE
International Symposium on Computer Aided Control System Design
Kohala Coast-Island of Hawai'i, Hawai'i, USA • August 22-27, 1999

WM1-3 2:40

Fuzzy-Neural Control with Application to a Heating System

SAMY MESBAH AND GRANTHAM PANG
DEPARTMENT OF ELECTRICAL & ELECTRONIC ENGINEERING
THE UNIVERSITY OF HONG KONG
POKFULAM ROAD,
HONG KONG
EMAIL: GPANG@EEE.HKU.HK
TEL: (852)-2857-8492
FAX: (852)-2559-8738

ABSTRACT

In this paper, a fuzzy-neuro approach will be presented for the design of bang-bang control system. A description of the architecture will be given. The multi-layered structure of the controller is essentially a neural network which resembles a fuzzy rule-based system. The robustness issue of an application to a heating system will also be described.

1. INTRODUCTION

The fuzzy-neuro approach for control applications has received a lot of attention recently. It is because it tries to combine the advantages of both fuzzy and neural approaches together. The fuzzy methodologies have been widely used as their representations are understandable and intuitive. However, the fine tuning of the performance of a fuzzy controller is difficult. On the other hand, the neural networks are more powerful in terms of their computational performance and learning capabilities, but weak from the viewpoint of intuitive representation. Hence, efforts to integrate the two approaches have been attempted by many researchers [1]. Many of these techniques are based on the equivalence of specific classes of rule-based fuzzy systems and the type of neural network proposed [2]. The idea is to construct a neural network that is functionally equivalent to a fuzzy system. Then, the neural network is trained to learn with more data, and the parameters are adjusted accordingly. The adjusted parameters can be shown as variations in the shape of membership functions of the original fuzzy system. Jang [3], Berenji [4], Horikawa [5], and Ishibuchi *et al.* [6] provide generalised methodologies for to a wide range of fuzzy systems. Yamakawa [7] has proposed an approach which is efficient but work for a more specific class of systems which contain only triangular complementary membership functions. All the above approaches have been successful in developing fuzzy-neuro controllers but share a lack of intuitiveness in the resulting neural network.

In this paper, the focus is on the use of a fuzzy-neuro approach for the bang-bang control of a system. These systems which rely on a controller with just the "on? and "off? states can be commonly found in many industries. For example, attitude control usually requires an on/off thruster [8] and flow control valve operated in a bang-bang mode [9]. Fuzzy control of a heating system has also been attempted with a bang-bang servo controller [10]. A set of fuzzy rules was used for the adjustment of the parameters of the servo controller.

2. FUZZY NEURAL NETWORK (FNN)

A fuzzy-neuro network based on [11] has been developed [2, 12]. The architecture of the FNN is consisted of three layers: fuzzification, rule reasoning, and defuzzification layers. These three layers are shown in Figure 1. Each layer is constructed and trained independently and then the three layers are connected afterward. This neural network structure is designed to mimic the fuzzy rulebase of an original fuzzy controller. The membership functions (input and output) and rules of an adjusted fuzzy controller can be retrieved from the FNN after training.

2.1. Fuzzification Layer. The first layer of the FNN performs the fuzzification. Let i be the number of inputs to the fuzzy controller. The fuzzification layer contains i sub-networks. Let x_i be the i^{th} input to the network. The i^{th} sub-network will have k outputs where k is the number of fuzzy membership sets corresponding to x_i. Each of the i sub-networks is trained individually with the membership functions corresponding to each x_i. Each resulting sub-network then produces a membership value μ_{ik} at each of its outputs for any crisp x_i. This array of sub-networks, which operate in parallel and are not connected to each other, perform the fuzzification of the controller inputs. It should be noted that each sub-network itself is a complete ANN and can have input and output layers and as many hidden layers as desired.

2.2. Rule Reasoning Layer. In this layer, the knowledge base which represents the fuzzy rules is represented. The knowledge is contained in the training of the rule reasoning sub-networks and in the structure (that is interconnections) between the fuzzification and defuzzification layers. The number of rule reasoning sub-networks is the same as the number of fuzzy rules in the original system. The number of inputs to each network corresponds to the number of preconditions in the rule it represents. The number of outputs is the number of consequents in the rule it represents. Each sub-network is again a complete neural network which can be trained to implement different kinds of conjunction operators.

2.3. Defuzzification network. The defuzzification layer is consisted of two sub-layers. The first one calculates fuzzy controls and the second sub-layer converts the fuzzy controls to a crisp value using the center of area method. The equations for the computations will be described later.

Fuzzy Controls Sub-layer. The number of inputs is the same as the number of fuzzy rules in the (that is rule reasoning sub-networks) of knowledge base. Let r be the number of rules. The output membership functions of the fuzzy rules are represented by weights in this part of the defuzzification layer. This representation requires the universe of discourse for the output to be discretized. If n is the number of discrete values chosen then define $U = \{u_1, u_2, \ldots, u_n\}$ as the set of discrete output values. The initial weights of this sub-layer are calculated with [2]

$$w_{ij} = \mu_{C_j}(u_i) \tag{1}$$

where $i = 1, \ldots, n$ and $j = 1, \ldots, r$. As well, C_j is the linguistic label of rule j and $\mu_{C_j}(u_i)$ is the degree of membership of C_j at $u = u_i$. The weights in this sub-layer are subject to change during learning.

Center of Area Sub-layer. The main purpose of this second sub-layer is to perform a center of area calculation and to produce a crisp output value. The weight of this layer, denoted by β_i, are calculated as follows:

Let u_i be the discrete values as defined in the last subsection.

$$\beta_i = \frac{u_i}{\gamma} \tag{2}$$

where

$$\gamma = \sum_{i=1}^{n} \mu_i \tag{3}$$

The crisp value u which results from defuzzification can then be computed with

$$u = \sum_{i=1}^{n} \beta_i \mu_i \tag{4}$$

The weights in this sub-layer are not affected by training. The reasons for this will become clear in the next section.

2.4. Training the FNN. This section deals with the training of the entire FNN with real input/output data. Error calculation and weight updating is described for each of the layers.

If u^d is the desired output value for an input set and u is the actual output generated by the FNN from that input set then the error function can be defined as

$$E = \frac{1}{2}(u^d - u)^2 \tag{5}$$

The center of area calculation must remain intact so the β are always calculated from the μ_i. The training algorithm does not change the β weights at all. The rule for updating the w_{ij} in the fuzzy controls sub-layer of the defuzzification layer can be derived from the gradient descent algorithm to be

$$w_{ij}(t+1) = w_{ij}(t) + \epsilon\alpha_j(u^d - u)(\beta_i - \sum_{l=1}^{n} u_l\mu_l/\gamma^2) \tag{6}$$

where ϵ is the learning rate and lies in $[0, 1]$ and α_j is the degree of firing of rule j, that is the j^{th} input of the defuzzification layer. The error terms which are propagated back to the rule reasoning layer can be shown to be

$$e_j = (u^d - u)\sum_{i=1}^{n} w_{ij}(\beta_i - \sum_{l=1}^{n} u_l\mu_l/\gamma^2) \tag{7}$$

The error at the input of the defuzzification layer is used to calculate the weight changes in the sub-networks of the rule reasoning layer. This can be done with the standard back propagation algorithm.

3. A Furnace Heating System

A model of the thermal characteristics of a typical house is given in [12, 13]. The plant model includes a furnace which has two states: "on" and "off", along with a simple model of the temperature change when the furnace is at either of these two "steady-state" operating points.

3.1. Nominal and Perturbed Plant Transfer Functions. The nominal plant is chosen to be as "average" as possible, that is an average sized house with a mid-size furnace and fairly good insulation. The extremes are chosen as just that: the fastest and slowest heating houses from the calculated parameter ranges. The nominal plant transfer function is chosen as:

$$P_{nom} = \frac{66.15}{(41.4s+1)(300s+1)} \tag{8}$$

and the extremes could be (in same order as above):

$$\tilde{P}_b = \frac{7.90}{(198.72s+1)(900s+1)} \quad \tilde{P}_t = \frac{321.16}{(5.51s+1)(120s+1)} \tag{9}$$

There is a very wide range of possible perturbations between the nominal plant and those at the extremes.

3.2. On/Off Switching Control Law. The on/off switching control law is the most common method used for the control of a system. This is widely used for temperature regulation due to its low cost and simplicity. The control logic can be specified as follows:

```
If (furnace off AND error > on threshold)
  Turn furnace on
Elseif (furnace on AND error < off threshold)
  Turn furnace off
End
```

It is found that the standard control method of on/off switching with simple provided acceptable results for the nominal plant but the performance was not at all robust. It should be noted that the above is a nonlinear control law.

4. Fuzzy Controller Design

A fuzzy controller for the heating system is described in this section. The choice of rules and membership functions was based mostly on intuition and rules of thumb, along with "tweaking" once acceptable performance was achieved. The simulation of the fuzzy temperature control system was accomplished with MEDAL [14], a MATLAB like software package which includes built-in fuzzy logic and expert system constructs.

4.1. Membership Functions and Fuzzy Rules. For the definition of the fuzzy rules, the temperature error together with the rate of temperature change are used to relate to the controller output. Linguistic variables are used for the description. The temperature error linguistic variable is associated with fuzzy terms called small, medium, and large. The temperature change linguistic variable is associated with fuzzy terms called negative, small, medium, and large. As for the output linguistic variable, The fuzzy terms on, mid, and off are used.

The mid membership function represents a case where it is not clear whether the furnace should be on or off. This was chosen to represent the decision a human operator would face when he or she is not sure of what to do. The defuzzified value will lie in $[0, 1]$ and is used to determine the value of $m(t)$ at each time step. Figure 2 contains plots of the membership functions used.

Below are examples of two fuzzy rules:

```
IF (TempErr is small AND DeltaTemp is negative)
  THEN Output is off
IF (TempErr is small AND DeltaTemp is small)
  THEN Output is mid
```

The operation of the fuzzy control system is as follows:

1. execute rules with current temperature error and temperature change from last time step
2. defuzzify output linguistic variable
3. if output ≥ 0.5, then the furnace will be on for the next time step, otherwise it will be off.
4. simulate system for next time step
5. return to step 1

A MEDAL m-file was written to perform the simulation. The results from the simulation are given in Figures 3 and 4. The result for $\tilde{P}_b$ shows that the furnace will be on for all time and yet the steady state temperature cannot reach the setpoint.

4.2. Results. With computer simulation, it has been shown that the fuzzy controller has succeeded to reduce overshoot in the output response. The "on" time is generally quite low as it is with on/off switching. For $\tilde{P}_t$, the furnace "on" times are very small and quite a bit of oscillation is still present. This suggests that the house heats up *very* quickly and that this is probably far too quick to be a realistic house. Nevertheless, the fuzzy controller does a very good job even with this impractical case. Even P_{nom} seems too quick to be a real house which suggests the model parameters need adjustment. This could be accomplished with experimental data but for the purposes of this example, it is sufficient to work with the derived models to show the usefulness of fuzzy control techniques in producing a fuzzy controller which works well with all the plant models.

5. Fuzzy-Neuro Controller Design

A fuzzy-neuro controller is described in this section. A comparison with the fuzzy control approach with a discussion on the robustness issue is also given. In this FNN, the fuzzification layer has two sub-networks which were trained with the membership functions for the input variables. Each sub-network has a single neuron input layer, two hidden layers with seven neurons each, and an output layer with three (temperature error) or four (temperature change) neurons. The output neurons correspond to the fuzzy terms: small, medium, large, negative membership values. Two hidden layers are used as it allows a reasonably complex mapping.

The twelve rules in the original fuzzy controller are implemented with twelve rule reasoning sub-networks in FNN. Only one sub-network was initially trained with a product AND function and its weights and thresholds copied to the other 11 sub-networks. These sub-networks are composed of two input layer neurons, two hidden layers each with seven neurons, and one output layer neuron. The choice of hidden layers is again arbitrary and was used to allow relatively complex functions to be represented by the sub-network. The sub-networks are initially trained to return the product of the two inputs (which are in the range $[0,1]$). Hence, the output is also in the $[0,1]$ interval.

In the defuzzification layer, the $[0,1]$ range of the universe of discourse of the output variable is divided into 21 discrete points. The weights are the membership values (μ_i) at each of the 21 discrete output values. It is interesting to note that with the 12 rules there are 12 sets of 21 weights in the fuzzy controls sub-layer and each of these sets can change independently. When the initial weight entry is made, there is much redundant information in the weights but this will change when the entire FNN is trained.

5.1. Comparison of the initial FNN and Fuzzy Controller. Ideally, the FNN should be equivalent to the pure fuzzy controller. There are several reasons why the results are not exactly the same. First, the input membership functions are smooth continuous functions in the FNN which differ slightly from the triangular membership functions defined in MEDAL. Secondly, the AND product is also trained in the FNN and is not an exact product function although it is very close. Small inconsistencies in these sub-networks can have a more profound effect on the entire system since they control the degree of rules firing. Thirdly, the output membership functions have been discretized and the center of area is thus an approximation based on these points as opposed to exact equations for trapezoids in MEDAL. Note that the network is not completely interconnected, and it is a very specific composition of sub-networks, numerical errors can be significant at the output of the controller. Figure 5 shows a comparison of the P_{nom} response with the original fuzzy controller and with the FNN before training.

5.2. Training of the Network. The objective of representing a fuzzy controller by an FNN was to tune the controller parameters with numerical data. The FNN, being a neural network, can be trained with the derived gradient descent algorithm. This allows the internal representations of the input and output membership functions and the definition of AND operator to be updated to accommodate the desired numerical data.

One main source of training data is from the actual case study simulations with the FNN. Any plant can be simulated and the data is saved to a file. The data can then be modified to control the heating system like a human operator. For example, the data can be changed so as to turn the furnace off a few time steps earlier to eliminate the overshoot. The structure of the neural network is such that very localised improvements can be made as long as "good" data is kept and included in the training set to ensure the "good" portions of performance are not changed. The formulation of fuzzy rules with appropriate membership functions has always been a problem in the design of fuzzy controllers. Even if an expert is available, it is difficult for him to put the necessary rules on paper. The advantage of training an FNN with case studies is that the controller can learn not to repeat undesirable controls much like a human operator.

5.3. Results. Here, the main objective of training is to reduce the overshoot in P_{nom} and $\tilde{P}_t$. Improvements have been made with the use of case studies data. As the fuzzy controller provided a very good response for P_{nom} and $\tilde{P}_t$, the network was trained to try and diminish the slight overshoot and control the regulation of $\tilde{P}_t$. The attempt involved taking case study data from the original FNN and updating the network to reflect a better control strategy. Figure 6 shows the results of simulation after this training has occurred.

The membership functions are practically identical to the original definitions except for the medium temperature change which has again intruded on the large domain. This time, though, the large membership function has not been significantly modified. This is not too surprising since case study data allows "fine tuning" of the system.

Now that a suitable FNN is available, some robustness testing is in order. The first test involves attempting the same simulations with different plants. Two more plant models were formulated with one between $\tilde{P}_b$ and P_{nom} and the other between P_{nom} and $\tilde{P}_t$. The equations are

$$\tilde{P}_{bn} = \frac{40}{(80s+1)(400s+1)} \qquad \tilde{P}_{nt} = \frac{150}{(20s+1)(180s+1)} \tag{10}$$

and the resulting simulations are shown in figure 7. These simulations show good responses and could further be improved with new case study training steps.

6. Conclusions

This paper has focussed on using fuzzy and fuzzy-neuro control philosophies for the design of bang-bang controllers. The capabilities of fuzzy and fuzzy-neuro control has been illustrated using a heating system. The model for house heating is based on a two state furnace. The standard control method of on/off switching has provided acceptable results for the nominal plant but the performance was not robust. Next, a fuzzy controller was designed and excellent nominal performance was obtained. The fuzzy controller is quite robust as well. Fuzzy control has been shown very promising for this application. Fine tuning of the fuzzy controller is desired and so the fuzzy controller is represented as a fuzzy neural network. This also provides a good starting point for the improvement of the fuzzy controller. The use of simulation data has been shown to be a valuable method for improving the performance of the network.

Both the fuzzy controller and FNN have met the performance objectives. Good tracking and regulation, and robust performance were observed. Both control strategies have produced results on the same order of energy consumption. This paper has shown that both the fuzzy and FNN control are powerful and very practical design methodologies.

References

[1] J.S.R. Jang, C.T. Sun and E. Mizutani. *Neuro-Fuzzy and Soft Computing.* Prentice Hall, 1997.

[2] Grantham K. H. Pang. Fuzzy-neuro method in control design. Emerging Computing Techniques for Engineering Design, NATO ASI Series, Vol. 149, pp. 204-221, Natplio, Greece, Springer- Verlag, 1994.

[3] J. S. R. Jang. Self-learning fuzzy controllers based on temporal back propagation. *IEEE Trans. on Neural Net.*, 3(5):714–723, 1992.

[4] H. R. Berenji. Learning and tuning fuzzy logic controllers through reinforcements. *IEEE Trans. on Neural Net.*, 3(5):724–740, 1992.

[5] S. Horikawa, T. Furuhashi, and Y. Uchikawa. On fuzzy modelling using fuzzy neural networks with the back-propagation algorithm. *IEEE Trans. on Neural Net.*, 3(5):801–806, 1992.

[6] H. Ishibuchi, R. Fujioka, and H. Tanaka. Neural networks that learn from fuzzy if-then rules. *IEEE Trans. on Fuzzy Systems*, 1(2):85–97, 1993.

[7] T. Yamakawa. A neo fuzzy neuron and its applications to system identification and prediction of chaotic behaviour.. In J. M. Zurada, R. J. Marks II, and C. J. Robinson, editors, *Computational Intelligence: Imitating Life*, pages 383–395. IEEE Press, 1994.

[8] Richard Y. Chiang and Roger J. Jang. Fuzzy logic attitude control for cassini spacecraft. *Proc. 3rd IEEE Conference on Fuzzy Systems*, pages 1532–1537, 1994.

[9] R. Ushiyama and H. Walden. Coldplate temperature regulation using fuzzy logic control. *SAE(Society of Automotive Engineers) Trans.*, 100(1):1790–1801, 1991.

[10] N.R. Sripada, D.G. Fisher and A.J. Morris. An application for process regulation and process control. *IEE Proceedings-D*, 138, 1987.

[11] F. Y. Wang and H. M. Kim. Implementing adaptive fuzzy logic controllers with neural networks: a design paradigm. to be published in *Journal of Intelligent and Fuzzy Systems.*

[12] Grantham K. H. Pang. Design of bang-bang controller based on a fuzzy-neuro approach. Proc. of 1996 IEEE Symp. on Computer-Aided Control System Design, pp. 1- 6, Dearborn, MI, U.S.A., September 15-18, 1996.

[13] W. Roots. *Fundamentals of Temperature Control.* Academic Press, New York, 1969.

[14] Grantham K.H. Pang. Medal: Matrix & expert system development aid language. *Proc. IEEE Symp. on Computer-Aided Control System Design*, pages 148–155, 1992.

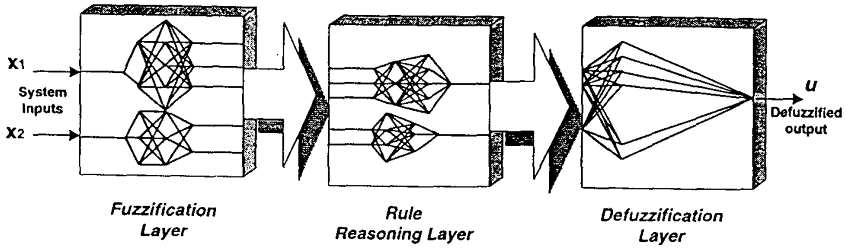
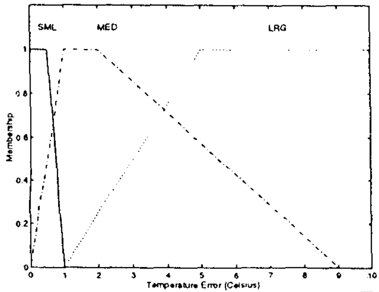
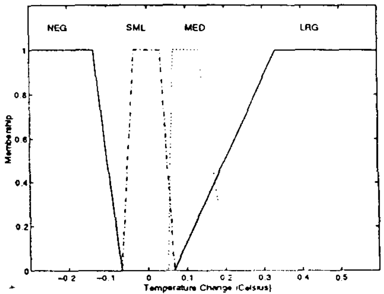


Figure 1: Structure of Fuzzy Neural Network (FNN)





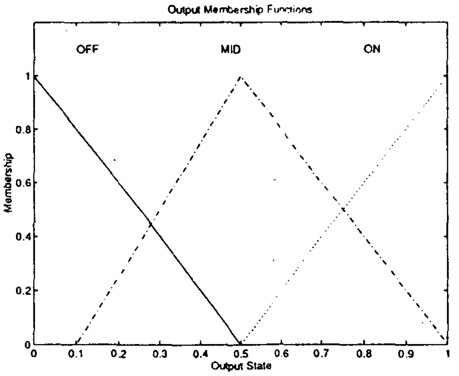


Figure2 : *Membership Functions for Fuzzy Controller*

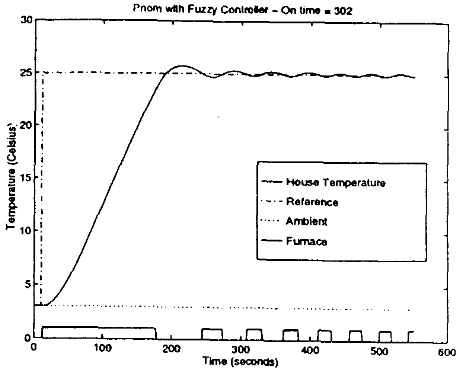


Figure 3 : P_{nom} *Response with Fuzzy Controller*

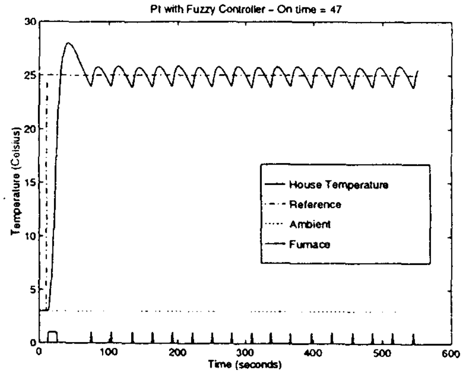


Figure 4 : $\tilde{P}_t$ *Response with Fuzzy Controller*

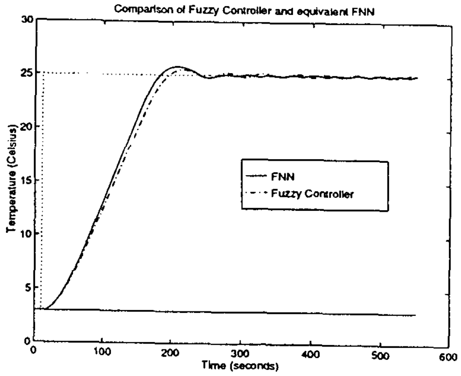


Figure 5 : P_{nom} *Response with Fuzzy controller and FNN*

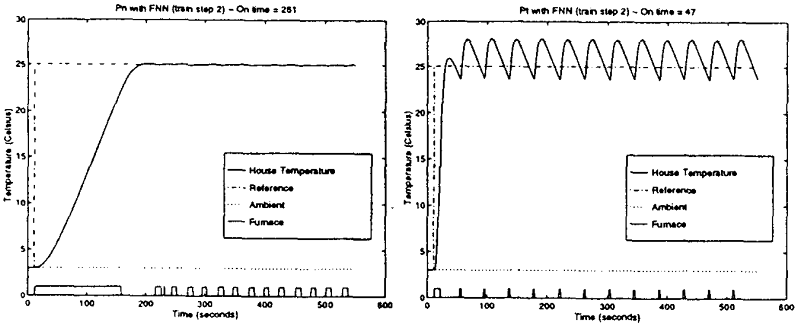


Figure 6 : P_{nom} *and* P_t *response after second training step*

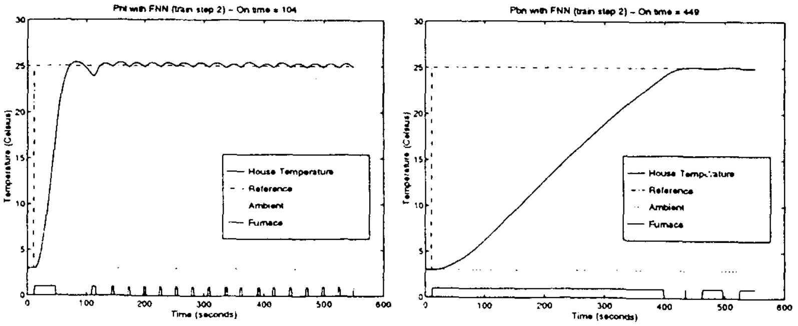


Figure 7 : $\tilde{P}_{bn}$ *and* $\tilde{P}_{nt}$ *responses*