A New Artificial Neural Network System
To Solve The ‘Ship Steering Problem’

M.Toygar Karadeniz
Unal Altinay

1 April, 1996

Introduction

The main purpose of the research going on ‘Intelligence Control’, is to build suitable intelligent controllers to leave some kind of plant(s) to their responsibility. The plant is usually something doing some important job which has strict constraints that are crucial to its proper operation. So, the controller which will take the control of that plant, must surely be robust and perfect. As, implementing and debugging such a controller on a real plant is too dangerous, some formulation of that controller is first implemented on some kind of simulation environment.

In this project proposal, the subject is to design an intelligent controller for steering a ship through some predetermined gates. This problem is known as ‘Ship Steering Problem’ and well studied before using neural networks. But, our aim, here, is to implement the new approach of the Takayuki Yamada and Tetsuro Yabuta by using their neural network training algorithms.

This project proposal paper consists of five main parts. Following this introductory part, the problem description is given, in detail. The next part tells about the neural network technique of Yamada and Yabuta which we intend to use in our implementation. Before the conclusion, we give a discussion about why we have chosen the mentioned technique and present the results of our brief survey about the subject.

Definition Of The Problem

‘Ship Steering Problem’ is one of the well known control problems, usually fed into new automatic controller designs to check and improve the system’s performance. The statement of the problem is simple that it is very easy to understand it and also visualize many of its extensions, at first sight. But to control it subject to its given constraints, is rather hard making it a good choice in the control simulation problem set.

The problem is about a ship wandering around in a x-y coordinate plane. There are various number of gates through which this ship is to be passed. The gates are all labeled and the order of passing through them, is important (Note that the last sentence tells about something that is not in the original problem statement. Labeling idea is preferred to reduce the search space and open to question.). The controller that is to be designed for this ship changes only the turning rate of the ship i.e. *r* in degrees/sec. But this change in the turning rate applies to the ship's orientation after a time lag. That is, mainly included into the problem just to simulate the outside effects like the water resistance etc.

The state of the problem is represented by four variables which can be summarized as *x,y* (current coordinates of the ship in the plane), θ (the orientation of the ship according to the vertical axis degrees), $d\theta/dt$ (the actual turning rate of the ship). Our controller designed for this problem, will take this variables as its inputs and produce only one output which is the control variable of the problem *r* (the desired turning rate of the ship). But this output is not acceptable without any constraint check. The main constraint in the case of *r*, is that $r < 15$ deg/sec i.e. the ship can not turn more than 15 degrees per second. In the initial state, the ship is at position 0.5 on the x-axis without any movement. The equations of movement are in accordance with the usual physical equations which are given as :

$$x(t+1) = x(t) + \Delta V \sin \theta(t)$$
$$y(t+1) = y(t) + \Delta V \cos \theta(t)$$
$$\theta(t+1) = x(t) + \Delta V \sin \theta(t)$$
$$d\theta/dt(t+1) = d\theta/dt(t) + \Delta(r(t)- d\theta/dt) / T$$

Also, there are some constraints concerning the ship which are *T* (time constant of convergence), *V* (constant velocity) and Δ (both sampling and control interval).

'Ship Steering Problem' is no imaginary problem which has no implications in the real world. Examining the steering delay because of external effects in this simulation environment leads to good solutions in the real implementation. Also the decision making in which orders of the gates are thought to be passed can greatly effect the future planning abilities of the real controllers (Note that, we have an idea to put a basis on this unconstrained order problem by constraining it with a suitable order, in our implementation.).

Various extensions to this problem can be discovered at first sight including the following :

- Many environmental effects like water turbulence and wind flows may be included into the system equations.
- Some set of the gates may be set hidden where some are visible in the first case. After the ship is passed all the ones in the first case, the new ones are presented.
- The coordinate plane may be filled up with various obstacles which prevent the ship going through some suitable paths.

As you may noticed, ‘Ship Steering Problem’ can be extended in many more ways. That is just the result of the reality concept concerned with this problem relative to our living world.

A Neural Network Controller For ‘Ship Steering Problem’

Among many papers about ‘Intelligent Controllers’, the one presented by Yamada and Yabuta seemed more promising in solving our ‘Ship Steering Problem’. The methodology and reasoning of the mentioned paper is outlined in this section.

The method described in Yamada and Yabuta, is applied to a learning-type direct controller for a predominantly linear, second order plant. The same method is going to be applied to the ‘Ship Steering Problem’. This method uses neural networks to benefit from both the flexibility and the learning ability of them. The neural network devised, has three layers with no inner feedback loop and no direct connections from input layer to output layer. Figure 1 shows the neural network model described above.

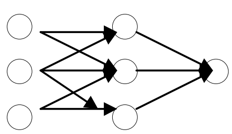

Figure 1

The authors stress on the fact that, many parameters of the neural networks, such as the optimum shape of the sigmoid functions (nonlinear saturated functions), are determined by trial and error, thus limiting the advanced applications of neural networks. To remedy this problem, an auto-tuning method for the sigmoid function shape, which will enable the neural network to be applicable to servo control systems, is proposed. It is proven in the paper that when the usual sigmoid function is used only in the hidden layer, sigmoid function tuning is the same as weight tuning. The mentioned usual sigmoid function; *g(x)* is expressed as the following equation :

$$g(x) = \{1-\exp(-x\ Y_g(P))\} / \{1+\exp(-x\ Y_g(P))\}$$

where x is the input of the sigmoid function, Y_g is the parameter determining its shape and P is the trial number. In these circumstances, another sigmoid function *f(x)*, is given in the paper, which can be expressed as:

$$f(x) = 2\ \{1-\exp(-x\ Y_g(P))\} / \{Y_g(P)\{1+\exp(-x\ Y_g(P))\}\}$$

The relation between the usual sigmoid function and the one proposed is as follows :

$$f(x) = (2 / Y_g(P))\ g(x)$$

Although, this sigmoid function is used only in the hidden layer too, it is proven that the sigmoid function tuning is not the same as weight tuning.

The auto-tuning method explained in the paper by Yamata and Yabuta, is going to be applied to the direct controller in order to confirm both its characteristics and practicality. The details of the auto-tuning method depend on the controller structure, which will be the subject of our study. Devising the motion equation is another benchmark, but for the time being let's assume that we have the motion equation of the controller simply as *Y(k).*

Then, the input vector I, of the neural network (all vectors in this paper being column vectors) is as follows:

$$I^T(k) = \{Yd(k+1),\ Y(k),\ Y(k-1),\ U(k-2)\}$$

where Y_d is the desired control value. As it can be seen from the equation, the output of the motion equations for the ship becomes, basically, the input to our neural network controller. By the way, the actual number of hidden units and the function in the output layer is going to be determined during the implementation of the controller, but it is for sure that the new sigmoid function of Yamata and Yabuta, is going to be used in the hidden layer. The neural network output which is the control value, is just the ship's input, *U*. So, if we summarize the last couple of sentences, the neural network control function can be expressed by the following equation:

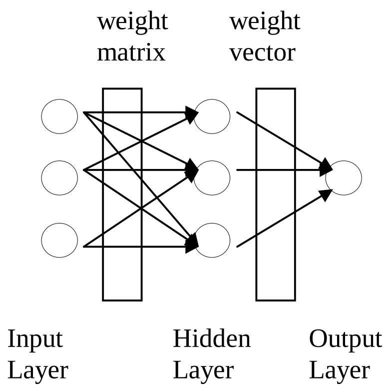

Figure 2

$$U(k) = f\{\ \omega^T(P)\ f(W(P)\ I(k))\ \}$$

Figure 2 shows the neural network scheme that is going to be applied to the controller.

Why Yamata and Yabuta Technique ?

In the case of information for determining our technique to attack the 'Ship Steering Problem', we did a brief survey on a couple of recent papers. Following lines give an idea about why we have chosen the Yamada and Yabuta technique but not the other ones presented in the papers that we have examined.

First paper which we had the chance to look at, was the Rolf Eckmiller paper on 'Neural Nets for Sensory and Motor Trajectories'. This paper is indeed a good source of information on the general concepts about trajectory controlling, but it lacks of the formal representations of the concepts. So, there is really no formulas that are to be implemented and it was not a good idea to concentrate on this paper.

The next paper was the Fu-Chuang Chen paper on 'Back-Propagation Neural Networks for Nonlinear Self-Tuning Adaptive Control'. It is mainly about self-tuning on tracking problems by using a back-propagated neural network.

The concepts seemed very suitable to our purposes on ‘Ship Steering Problem’ at first sight. But when we examined the technique proposed there, it became clear that the formulas derived, are not much suitable for us. In that paper, the controller is designed by coordinating two distinct neural networks and joining them by an output node. So, the neural network function becomes something like :

$$y = f(...) + g(...)u$$

where *f(...)* and *g(...)* are two non-zero functions each being approximated by one of the two neural networks and *u* is the input to the system. As you may have guessed, this function structure is totally different from the one to be used in our implementation as the *r* function. But the idea of using *n* distinct neural networks for different purposes gave some insight to us for sure (i.e. using a mixture of experts structure or like).

The last paper of our brief survey is the Q. H. Wu, G. W. Irwin and O. W. Hogg paper on ‘Neural Network Regulator’. The title seemed to be very suitable for our current research, but we soon recognized that this paper gives information on some basic concepts like ‘What is a MLP and what does BP algorithm?.’ and the described neural network regulator is nearly equivalent to a simple perceptron training. As we are looking for new and interesting approaches on intelligent control by neural networks in this project, we did not spend so much time on this paper, either.

After examining all the mentioned papers closely, we decided to concentrate on Takayuki Yamada and Tetsuro Yabuta paper on ‘Neural Network Controller Using Auto-tuning Method for Nonlinear Functions’. This paper and the technique which we intended to use, is fully described in part III.

Conclusion

In the first place, our controller design is still at a premature state, as this paper gives only the proposal. In fact, the design have not been tried on any examples yet. So, everything described is a good estimation for an implementation rather than the implementation itself. There may be many loose points as well as totally wrong assumptions. But this particular paper is good at describing the problem and giving an idea about the intended solve technique. Surely, the loose points of the design will be the subject of our effort.

References

[1] T. Yamada and T. Yabuta, “Neural Network Controller Using Auto-Tuning Method for Nonlinear Functions,” IEEE Transactions On Neural Networks, Vol. 3. 4 July 1992, pp. 595-601.

[2] R. Eckmiller, “Neural Nets for Sensory and Motor Trajectories,” IEEE Control Systems Magazine.

[3] F. Chen, “Back-Propagation Neural Networks for Nonlinear Self-Tuning Adaptive Control,” IEEE Control Systems Magazine July 1990, pp. 44-48.

[4] Q.H. Wu, G.W. Irwin and B.W. Hogg, “Neural Network Regulator,” IEEE 1991 Control Conference Pub. 332 Vol. 1, pp. 145-150.