LEARNING TO BALANCE THE INVERTED PENDULUM USING NEURAL NETWORKS

Victor WILLIAMS and Kiyotoshi MATSUOKA
Department of Control Engineering, Kyushu Institute of Technology
Sensui 1-1, Tobata, Kitakyushu. 804 Japan.

ABSTRACT - Neural networks can be used to control nonlinear unstable systems. In this paper a learning architecture is proposed for training a neural network controller to provide the appropriate force to balance an inverted pendulum. In the present approach the control rule is given on the output space of the inverted pendulum introducing a priori knowledge in the learning algorithm. The system uses two neural networks, one for the identification and the other for the controlling. The identification network learns to identify the plant dynamics while the controlling network simultaneously changes its characteristics using the result of the identification to control the actual plant. The training process is performed using the back-propagation learning algorithm. The inverted pendulum system is simulated to illustrate these ideas. The controller is able to balance the inverted pendulum and guide the cart to the center of a track, demostrating the effectivity of the learning method.

1. INTRODUCTION

As an example of the neural control of nonlinear unstable systems, this paper addresses the balancing control of the inverted pendulum. The inverted pendulum has been used as a useful laboratory idealization of unstable mechanical systems and at the same time it is also a good example when considering the application of neural networks to control systems.

Several neural network architectures have been proposed for controlling the inverted pendulum. For example, the control with forward modelling was studied by Jordan and Jacobs [2]. In this approach the control system learns a model that relates the current state of the plant and the current control signal to a prediction of future failure.

Also the learning control of the inverted pendulum by means of a neuro-controller was proposed by Kitamura and Saitoh [4]. In this approach the system is composed of a neural controller, a generator of the desired output and a evaluator. In the generator of the desired output at the next time step, given the position and velocity of the cart. In order to move it to a specified point, the angle and angular velocity of the pendulum are generated by a pair of equations in which they took into consideration a-priori knowledge of the behavior of the pendulum. The evaluator is used to decide if the output of the controller is right or wrong and depending on the case, generate the teacher signal for the training of the neuro-controller, based on the difference between desired value and control output.

In this paper we propose a different approach in which the control rule is given on the the output space of the inverted pendulum introducing a priori knowledge in the learning algorithm, because, in general, the konowledge can be given on the state variables space, in a relatively simple manner.

CH 3065-0/91/0000-0214 $1.00 © IEEE

2. INVERTED PENDULUM

The inverted pendulum consists of a movable cart and a pole mounted on it with a pivot having no friction, as shown in fig. 1. The cart is allowed to move within the bounds of a one-dimensional horizontal track and the pendulum is free to fall around the pivot. The objective is to balance the pole and guide the cart to a specified position on the track.

The dynamics of the system obeys the following nonlinear differential equations.

$$d\Theta/dt = \omega \tag{1}$$

$$d\omega/dt = g_1(\Theta,\omega,f) = \frac{g\sin\Theta - \cos\Theta\,\dfrac{(f + m_p l\omega^2\sin\Theta)}{m_c + m_p}}{\dfrac{4l}{3} - \dfrac{l m_p \cos^2\Theta}{m_c + m_p}} \tag{2}$$

$$dx/dt = v \tag{3}$$

$$dv/dt = g_2(\Theta,\omega,d\omega/dt,f) = \frac{f + m_p l\{\omega^2\sin\Theta - (d\omega/dt)\cos\Theta\}}{m_c + m_p} \tag{4}$$

where

f: force applied to the cart's center of mass,
Θ: angle of the pendulum with respect to the vertical,
ω ($=d\Theta/dt$): angular velocity of the pendulum,
x: horizontal position of the cart relative to the track,
v ($=dx/dt$): horizontal velocity of the cart,
g ($=9.8m/s^2$): acceleration due to gravity,
m_c ($=0.01Kg$): mass of the cart,
m_p ($=0.1Kg$): mass of the pole,
l ($=0.5m$): half of the pole length.

In the computer simulation described in § 5, the above differential equations are transformed into the following difference equations with a time step of T=0.01s, using the Euler approximation.

$$\Theta(t+T)=\Theta(t)+\omega(t)T \tag{5}$$
$$\omega(t+T)=\omega(t)+g_1(\Theta(t),\omega(t),f(t))T \tag{6}$$
$$x(t+T)=x(t)+v(t)T \tag{7}$$
$$v(t+T)=v(t)+g_2(\Theta(t),\omega(t),g_1(\Theta(t),\omega(t),f(t)),f(t))T \tag{8}$$

These equations are assumed to be not known; they were only used to simulate the dynamics of the plant.

3. LEARNING THE MODEL

In order to determine the appropriate forces to balance the pole at the specified position it is needed to know the dynamics or to learn a model of the inverted pendulum. Learning a model of the inverted pendulum is basically a system identification problem. Here it is done by a multilayer neural network capable of modeling nonlinear plants.

Fig. 2. shows a schematic diagram of the whole system. It consists of the plant (the inverted pendulum) and two neural networks, one (right in the figure) for identification and the other (left) for control. The neural networks are both standard layered ones using the following sigmoidal activation function.

$$h(z)= [1-\exp(-z)]/[1+\exp(-z)]. \quad (9)$$

The neural identifier consists of four output units and three hidden units. The state variables, θ, ω, x and v, observed from the plant and the control force, f, provided by the controller are applied to the network as inputs. The hidden units receive these variables, and the output units receive them as well as the outputs of the hidden units. Here we are assuming that the states of the plant are directly observable without any kind of external interference. Four outputs of the network represents the prediction of the state variables of the inverted pendulum at the next time step.

The neural network is trained using the back-propagation learning algorithm such that the output of the network closely matches the output of the real plant. The training process begins with an initial state which is generated randomly in each run. At time t, during each run, the input of the neural network is set equal to the current state of the plant. The neural network is trained, using the value of the next state of the plant as a desired response, so as to predict the next state of the plant at time t+T.

At each sampling time, the calculation is performed in three steps. In the forward propagation step, the output of each processing unit in the two neural nets is calculated layer by layer from input to output. In the backward propagation step the error between the target output and the actual output is calculated, and it is propagated backward through the network only in the neural identifier. The weight modification takes place after the backward propagation step is completed. The weight of all interconnections between units are modified using the output of each unit and the backpropagated error.

4. LEARNING THE CONTROLLER

The neural controller is composed of four input units (corresponding to the state variables), one output unit (producing the control signal), and three hidden units. It contains direct connections from input to output. At each sampling time, the neural controller receives four state variables of the plant. Employing this information, the neural controller determines the force necessary for maintaining the pendulum vertical and the cart moving toward the center of the track.

The objective of the learning is to teach the controller how to balance the pendulum, but the problem here is what kind of signal should be given as teaching signal. In general, when neural networks are employed in control applications, it becomes a problem to decide what kind of signal have to be generated as teacher. Here, we utilize certain a priori knowledge suggesting which kind of control signal should be applied to balance the pendulum at a specified position.

When considering the introduction of such kind of a priori knowledge it is possible to take into account two alternatives. In the first line the control rule is directly specified on the force. This approach was employed by Kitamura and Saitoh as was mentioned in the Introduction. In the second line the knowledge is described on the state space of the plant, and the control rule specifying the force in terms of the state variables of the plant is indirectly derived from it. Here, we adopt the second approach because, in general, the knowledge is given on the state variables space with less difficulty.

The a priori knowledge considered here is as follows:

(K1) when the pendulum is falling from the vertical position and we change the angular velocity of the pendulum by certain amount in the opposite direction on which the pendulum is falling, the pendulum will be forced to move in the direction of the angular velocity.

(K2) when the cart is moving at a certain distance from the center of the track and the pendulum is vertical, the pendulum must tend to fall in the direction of the center of the track. However, if the cart is moving to the right for example, then the pendulum must tend to fall is the opposite direction (left). If it is moving to the left the pendulum must tend to fall in the opposite direction (right).

Utilizing a combination of the above mentioned parameters, the desired target value $\omega^d(t+T)$ of the angular velocity of the pendulum at time t+T, can be written specifically in equation form as follows.

$$\omega^d(t+T) = 2h(-5(\theta(t)+ 0.1x(t)+0.2v(t))) \tag{10}$$

Where h is the sigmoidal function defined by (9). As for the other state variables, the targets are not specified. It should be noted that the objective of the learning is described on the state space which is generated by the force, not on the force itself.

The backpropagation learning can also be used for training the controller. The training consists of three steps. In the forward propagation step, the output of each processing unit in the neural controller and idendifier are calculated in the forward direction; it is the same as in the last section, not requiring to repeat the calculation. In the backward propagation step the error between the target output specified by the equation (10) and the actual output at the actual angular velocity of the pole is calculated, and it is applied to the corresponding output unit of the neural identifier and is propagated backward through the neural identifier to the neural controller. The modification of the connection takes place only in the neural controller, in the same way as the learning in the neural indentifier described in the last section.

5. SIMULATION RESULTS

The cart was allowed to move in an interval equal to five times the length of the pole. The pole was free to fall from a vertical position (zero degree) to plus or minus sixty degrees.

Each run started from an initial state in which the state variables are chosen at random and ended when the pendulum or the cart surpassed the limits mentioned before or a key was pressed on the keyboard. The weigths of the connections in the two neural networks are initialized at random with values ranging from -1 to 1.

Fig.3 shows the simulation results. The three curves illustrated in the figure represents the response of the inverted pendulum controlled by the neural controller at three times during learning. In order to evaluate the effectivity of the learning, we stopped the learning process temporarily after the 10th, 15th, and 20th runs and observed the responses of the pendulum utilizing the same initial values.

Fig.3.a. shows the curves corresponding to the angle of the pendulum and fig.3.b. shows the output curves corresponding to the position of the cart. In the early stage of the learning (dotted and chained lines), the

target values were not achieved. At the 20th run (solid line) the angle of the pendulum and the position of the cart change considerably initially, but gradually the desired values of the angle and position are reached and the pendulum is balanced for a long period of time, while the cart is positioned at the center of the track. Clearly the performance improves with learning.

After the learning is finished, if the external disturbance is added, the controller is able to restore the pendulum to the vertical position (zero degree) after the cart moves back and forth two or three times.

6. CONCLUSION

The neural identifier was able to represent the dynamics of the inverted pendulum. Nonlinearity in the model was essential for accurate modeling of the dynamics.

Controlling the nonlinear kinematics of the inverted pendulum required a nonlinear controller, implemented by another layered neural network. The learning is done in a layered neural network using the back-propagation method.

Introducing a priori knowledge proved to be useful in the implementation of the learning algorithm, in which the control rule was given in the state variables space generated by the force, not on the force itself.

Simulation results showed that the learning algorithm performed very well, requiring only a few tens of runs to train the neural controller to balance the pendulum successfully while the cart was guided to the center of the track.

REFERENCES

(1) A.G. Barto, R.S. Sutton, and C.W. Anderson: Neuronlike adaptive elements that can solve difficult learning control problems. IEEE Transactions on System, Man, and Cybernetics, smc-13(5),1983.

(2) M.I. Jordan and R.A. Jacobs: learning to control an unstable system with forward modeling.

(3) D.E Rumelhart, G.E. Hinton, and R.J.Williams: Learning internal representations by error propagation. In Rumelhart and McClelland (eds.), Parallel Distributed Processing, vol. 1, chap. 8, MIT Press, Cambridge, MA, 318-362, 1986.

(4) S. Kitamura and M. Saitoh: Stability Learning Control of the inverted pendulum using neural networks (in Japanese). Knowledge and Intelligent systems simposium. pp. 61-64. March, 1990.

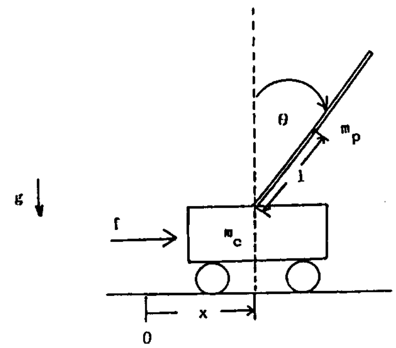

Fig. 1: Inverted pendulum.

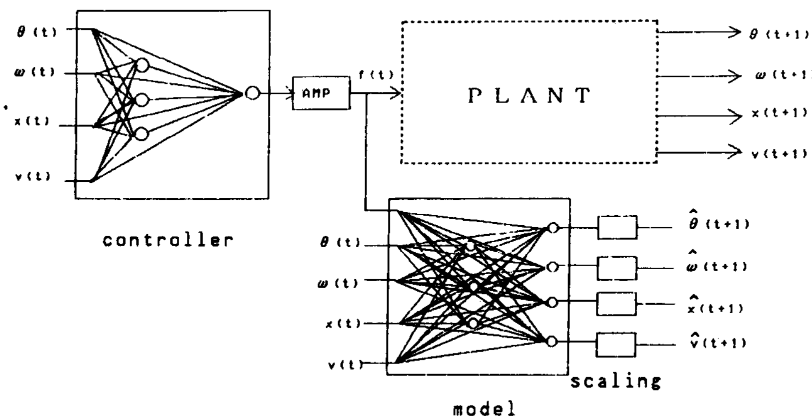

Fig. 2: Whole System