A Cartpole Experiment Benchmark for Trainable Controllers

Shlomo Geva and Joaquin Sitte

It is widely believed that balancing an inverted pendulum is a difficult nonlinear control task. Many researchers have used a variant of the inverted pendulum problem, the cartpole, for demonstrating the success of their neural network learning methods. It has been known for a long time that a linear control law, implemented by a single artificial neurone, can control the cartpole. Not noted before was that a random search in weight space can quickly uncover coefficients (weights) for controllers that work over a wide range of initial conditions. This result indicates that success in finding a satisfactory neural controller is not sufficient proof for the effectiveness of unsupervised training methods. By analyzing the dynamics of the linear controller, the cartpole problem is reformulated to make it a more stringent test for neural training methods. A review of the literature on unsupervised training methods for cartpole controllers shows that the published results are difficult to compare and that for most of the methods there is no clear evidence of better performance than the random search method.

Inverted Pendulum Problem

Almost 30 years ago Widrow & Smith 1963 [1] showed that a single McCulloch-Pitts type artificial neurone can control a cart to balance a pole mounted on it. Still, training a neural net to control a cartpole experiment is considered a difficult problem and it has become a popular test for neural network learning algorithms ([2]-[6], [8], [10]-[17]). Several factors contribute to the popularity of the broom balancing, or inverted pendulum problem: It is simple to describe and easy to understand, and although most of us can balance a broomstick with a little practice, it looks like a difficult problem. Curiously enough, humans seem to be unable to balance the cartpole although they may easily balance a pole of the same length on the palm of their hand. The inverted pendulum is a textbook example of an inherently unstable control system. As such it has been analyzed in detail with conventional control theory providing a convenient reference for assessing neural network controllers [9]. Also, a broom balancer can be built with moderate effort making a vivid demonstration. Finally, because it is a real-time problem there are constraints on the response time of the controllers. Controllers that work on computer simulations but are too slow for real time implementation are of purely academic interest, if at all.

Humans can learn on their own to balance a pole. Neural networks can be trained by a teacher to balance a pole. Is it then possible to design neural network controllers that also learn the task on their own? Training schemes for cartpole controllers that do not need a teacher (unsupervised learning) have to solve a *credit assignment problem*: Rating the influence of each individual control action on the outcome of a sequence of actions. Here we argue that despite attempts by several researchers to find a solution to the credit assignment problem for the inverted pendulum, clear evidence for their success is still missing. From the evidence available, most learning methods designed to deal with credit assignment in the cartpole problem do not seem to perform better than a random search in parameter (weight) space. In this article we show that it is easy to find, by random searches in weight space, single neurone controllers that achieve the fundamental control objectives of maintainig the pole upright and bringing the cart to the center of the track. The controllers found in this way vary widely in *quality*. Below, we refer loosely to controllers that achieve the fundamental control objectives as *good* controllers. It is necessary to discriminate between good controllers, therefore we introduce quality criteria and propose a benchmark specification for cartpole experiments that will make it very unlikely to find successful neural controllers by chance. Such a benchmark provides a focus for the research on unsupervised learning of real-time control tasks and makes it possible to compare the results obtained by different researchers. Comparison of the results of past research on the cartpole control is very difficult, if not outright impossible, because researchers did not adhere to a uniform specification of the learning task. It also seems that there is still widespread misunderstanding of the cartpole learning task.

The purpose here is to present a thorough analysis of the cartpole problem, to clear up implied or explicit misconceptions contained in earlier work, and to propose a set of conditions to make it a useful and well-defined benchmark for neural network training algorithms.

In this article, first we describe the cartpole problem in its most widely used form. Then we analyze control laws that are linear in the state variables of the cartpole, for both *bang-bang* and proportional control strategies. We show how likely it is to obtain succesful controllers by random search and characterize the performance of the controllers obtained in this way. There are large differences between the two control strategies that will affect training methods. We then formulate our benchmark proposal and conclude with a review and some observations on past work on neural network controllers for the cartpole experiment.

Cartpole Experiment

The cartpole is a special form of balancing a broomstick. The most often used version of the cartpole balancing problem is the one described in 1983 by Barto, Sutton & Anderson [4]. The lower end of a pole is mounted on a cart in such a way that the

The authors are with the School of Computing Science, Queensland University of Technology, GPO Box 2434 Brisbane, Q 4001 Australia. Email: s.geva@qut.edu.au or j.sitte@qut.edu.au.

0272-1708/93/$03.00©1993IEEE

pole can only swing in a vertical plane parallel to the direction of motion of the cart. To balance the pole the cart is pushed back and forth on a track of limited length. Balancing *fails* when the inclination of the pole exceeds preset limits, or when the cart hits the stops at the end of the track. The aim is to find a controller that prevents the cartpole from failing. A more demanding version of the cartpole experiment requires the controller to balance the pole and bring the cart back to the center of the track. Even on a track of limited length avoiding failure does not imply centering. There are controllers that avoid failure but do not center the cart while others would do a very good job of balancing and centering if they were allowed to run on a slightly longer track.

The state of the cartpole system is described by four variables: the position x of the cart, its velocity v, the pole angle θ, and the angular velocity ω. The force applied to the cart provides the controlling action. Fig. 1 shows the conventional definitions of x and θ.

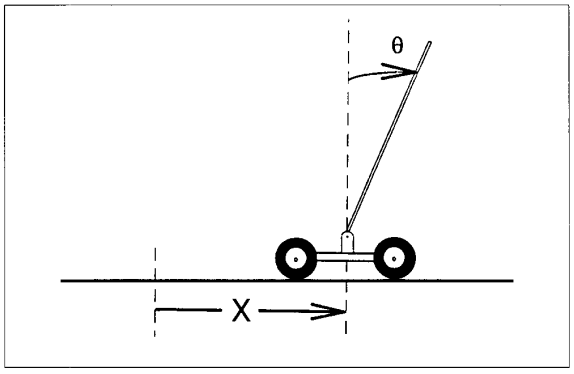

Fig. 1. The cartpole: definition of cart position x and pole angle θ.

Most of the published works on neuromorphic controllers for the cartpole use computer simulations of the dynamics of the system.

The dynamic equations for the frictionless cartpole are

$$\frac{d^2\theta}{dt^2} = \frac{g \sin\theta - a\cos\theta - \mu_p \omega^2 l \cos\theta \sin\theta}{l\left(\frac{4}{3} - 3\mu_p \cos^2\theta\right)} \quad (1)$$

$$\frac{d^2x}{dt^2} = \frac{\frac{4}{3}a + \left(\frac{4}{3}\omega^2 l - g\cos\theta\right)\mu_p \sin\theta}{\frac{4}{3} - 3\mu_p \cos^2\theta} \quad (2)$$

where μ_p is the reduced mass of the pole

$$\mu_p = \frac{m_p}{m_p + m_c} \quad (3)$$

and a stands for $F/(m_p + m_c)$. The standard values for the parameters used in the simulations are those given by Barto *et al.* [4] and are reproduced here in Table I.

Table I
Standard Cartpole Parameters

track limits	$\pm 2.4m$
failure angles	$\pm 12°$
gravity (g)	$-9.81\ m/s^2$
length of pole ($2l$)	$1m$
mass of the cart (m_c)	$1.0kg$
mass of the pole (m_p)	$0.1kg$
magnitude of control force (F)	$10.0N$
intergration time step	$0.02s$

Except for the magnitude of the control force, the particular choice of the other parameters has little effect on the nature of the control problem. It follows from the dynamic equations (1) and (2) that the effect of choosing a light pole on a heavy cart is to reduce the contribution of the more complex nonlinear terms. Simulation results show that the mass of the cart has little qualitative effect on the solution to the problem apart from increasing the magnitude of the necessary control force proportionally. The equations of motion given by Barto *et al.* [4] include small friction terms. The small values given to the friction coefficients make their inclusion largely cosmetic in that it adds a touch of some realism to the formulation of the problem. However, friction may help to stop the cart in cases where the controllers would allow the cart to slowly drift away or oscillate. If we think of the cartpole as a sensory motor control task in a biological system rather than an engineering control problem we would rather consider a control acceleration directly delivered to the base of the pole without the intermediary of a cart. The choice of the control variable is somewhat arbitrary. For example, for the control of a real cartpole system we would more likely to be concerned with the delivery of a current to the motor driving the cart. The most reasonable thing to do is to avoid the question of delivery of the control force altogether and just assume that we have a mechanism capable of applying the desired force to the base of the pole. In much of the earlier work on the cartpole the control force was taken to be of constant magnitude. Control could only be exerted by alternating the direction of the applied force. This control scheme is known as *bang-bang* control. In computer simulations the control force is usually allowed to change at every time step of the numerical integration. More recently a control force of variable magnitude has been tried emulating continuous force control. A continuously variable control force gives better control but it has to be kept in mind that in practice there will always be a maximum force that can be delivered by a motor. The quality of a controller is measured by the range of initial conditions from which it will stabilize the cartpole, the time taken to center the cart and the amplitude of any residual oscillation or displacement. Time to failure is often used to characterize controllers that cannot balance the pole indefinitely.

Linear Control Law

Widrow and Smith [1], [3] used the δ-rule to teach a single, McCulloch-Pitts type neurone to control the cartpole with the bang-bang technique. Their teacher signal was generated by a

linear classifier that divided state space into two regions. The output of the classifier determined the sign of a fixed magnitude control force:

$$F = k\ sgn\ (w_\theta\theta + w_\omega\omega + w_x x + w_v v) \tag{4}$$

The state of the cartpole was sampled at regular intervals and classified. The result of the classification determined the sign of the fixed force, of magnitude k, that was applied to the cart for the duration of the sampling interval. For the linearized dynamic equations it can be shown that the linear control force

$$\vec{F} = k(w_\theta\theta + w_\omega\omega + w_x x + w_v v) = \vec{k}(\vec{w} \cdot \vec{s}) \tag{5}$$

minimizes the quadratic error measured by the time integral of the square of the four state variables and the control force [30].

A qualitative analysis of the linear control law reveals that all the weights have to be positive. The analysis also helps to understand the control strategy embodied in the linear control law. Consider first the situation where the cart is stationary at the center of the track, and the pole is leaning at a positive angle with no angular velocity. The control action is then determined by:

$$\vec{F} = kw_\theta\theta \tag{6}$$

Clearly in this case only a positive force will erect the pole, and hence w_θ has to be positive. Similarly, if the cart is stationary at the center, with zero pole angle, but with positive angular velocity then clearly only a positive w_ω will produce a force that reduces the angular velocity.

That the weights w_x and w_v also have to be positive is not so obvious. Suppose that the cart is stationary somewhere on the right hand side of the track with the pole perfectly balanced. In this case the control force is determined by w_x. A positive weight will cause the force to accelerate the cart away from the center of the track. This action will initially move the cart further away from the center, and may appear to be incorrect. However, as a result of this action the pole will start falling to the left, making θ and ω negative. As the angle and the angular velocity become more negative their negative contribution to the force (equation (5)), attempting to erect the pole, will overcome the positive contribution from x. The net effect, over time, of the spoiling effect of w_x, and correcting effect of w_θ and w_ω is to accelerate the cart towards the center. To see how this net effect comes about consider the pole in a stationary upright position. A sequence of n control actions in one direction will cause it to fall in the opposite direction, helped by gravity. More than n control actions in the reverse direction are required to restore balance. This is to compensate for the work done by gravity during fall and the pull of gravity during recovery. Hence, the cartpole receives a net acceleration in the direction opposite to the initial direction of force application.

Being accelerated towards the center the cart eventually overshoots and the opposite sequence of control actions will tend to bring it back towards the center again. This mechanism alone will cause the cart to oscillate about the center. Without friction the oscillations will not dampen with time. To stop the cart at the center of the track, a mechanism is required that will take the role of friction. That is exactly what a positive weight w_v provides. Suppose that the cart is at the center of the track, with the pole balanced, but moving with constant velocity to the right. The braking is induced by a positive w_v, through the angular contributions, in the same way as centering is induced by a positive w_x. The velocity contribution to the force causes the cart to accelerate to the right, thus increasing the cart's velocity. However, again the side effect of this action is to cause the pole to fall to the left. The control actions that follow to erect the pole produce the desired net result of slowing down the cart.

Now we can appreciate the strategy embodied in a linear controller: The weights $w\theta$ and $w\omega$ work towards maintaining the pole in an upright position. The weight w_x indirectly causes the cart to accelerate towards the center of the track, by causing the pole to lean in that direction. The weight w_v indirectly slows down the cart, by causing the pole to lean in a direction opposite to the direction of movement. The relative magnitudes of the weights are such that balancing takes priority over centering and braking.

The interplay of pole angular movement and cart movement also illustrates the extent to which the requirement of centering adds to the credit assignment problem. Controllers that avoid failure but do not center only solve a much weaker form of credit assignment.

Armed with this understanding one can now see that some apparently logical control strategies are actually wrong. For example, it seems reasonable to expect that any controller that can balance the cartpole at some position can also be induced to center it, by adding a slight bias, proportional to x, to the inclination reading [19]. The reason is that the controller is made to see the pole inclined slightly to the center when in fact it is balanced. The argument then says that the controller will push the cart towards the center in an attempt to restore balance. This is contrary to our previous analysis that requires initial application of force in opposite direction. More precisely, it is wrong because it produces a term with negative weight in the linear controller, and we already know that for a linear controller to balance the pole and center the cart all its weights have to be *positive*. The effective angle θ_e sensed by the controller is the real angle θ minus the bias αx ($\alpha > 0$):

$$\theta_e = \theta - \alpha x. \tag{7}$$

For a linear controller, $w\theta$ has to be positive, therefore the bias term generates the contribution

$$w_\theta\theta_e = w_\theta\theta - w_\theta\alpha x \tag{8}$$

with the last term being a *negative* linear term in x.

The linear control law obviously fails when the cart is far from the center of the track. For large enough $|x|$ the term w_x in the control law (5) dominates all other terms, resulting in a large force always being applied in the same direction, and that inevitably leads to failure. The easy solution to that problem is to limit the position input signal, by sending it through a saturable amplifier, that is, a single input sigmoidal neurone. The solution can be equally applied to large velocities.

Random Searches in Weight Space

Surprisingly, by understanding the control strategy it is easy to pick, by trial and error, a set of weights that balances the pole and centers the cart. This raises the possibility that a random search in weight space might also be effective.

To test this hypothesis we generated 10 000 unitary weight vectors with random orientations. Linear controllers with these weight vectors were tested in a computer simulation for their ability to prevent the cartpole from failing within the first 300 s after release from various initial conditions. For the computer simulation we used the parameters from Barto *et al.* [4] listed in Table I and frictionless dynamics. The control force was updated at every integration time step of 0.02 s (50 Hz sampling). The controllers were tested in the bang-bang and the continuous force control mode. We made the assumption that the maximum force deliverable by the motor is limited to 10 N with $k = 50$. Because we know that the weight vectors have to have positive components we used a second population of 10 000 vectors chosen from the positive quadrant. The results in Table II show that one out of twelve random weight vectors could balance the pole for at least 5 min in the bang-bang regime when the cart was released from the center of the track and with the pole in equilibrium. For continuous force this initial condition is trivial since the control force is always zero and there is nothing in the simulations to break the unstable equilibrium.

Table II
Number of Controllers with Random Unitary Weight Vectors that Prevent the Cartpole from Failing within 5 min when Released from Various Initial Configurations

start state	bang-bang		continuous force	
$(x,\dot{x},\theta,\dot{\theta})$	all	positive	all	positive
0, 0, 0, 0	833	906	10,000	10,000
1, 0, 0, 0	31	344	8	110
0, 1, 0, 0	13	241	8	98
0, 0, 0.17, 0	21	352	8	110
0, 0, 0, 1	7	182	4	80
1, 1, 0.17, 1	2	52	0	30

Releasing the cart from the center of the track with the pole in equilibrium is a very forgiving initial condition for bang-bang control. Almost as many controllers from the totally random population pass the test as from the positive quadrant population. The explanation for it is that controllers with small negative w_x and w_v will survive the first 5 min, although they will not stay at the center but rather oscillate or drift away slowly. When the initial condition is made slightly more difficult by releasing the cart at 1 m to the right of the center, the controllers of dubious quality no longer pass the test. The ratio between those from the positive quadrant to those from the totally random population comes close to the expected ratio of 16. When the knowledge that the weights have to be positive is discarded (search over all orientations) controllers that pass the test from a difficult initial position become hard to find. The last row in Table II corresponds to such a condition: The cart is released at 1 m to the right of the center, moving right at 1 m/s from an angle of 0.17 rad (approx. 10°) and falling with an angular velocity of 1 rad/s (approx. 57°/s).

The results clearly show that it is easier to find bang-bang controllers that pass the test than continuous force controllers. In turn, as will be shown later, the continuous force controllers are more robust and more precise than the bang-bang controllers.

Using Time to Failure to Rate Controllers

The purpose of a learning method is to do better than a simple random search. Even if a learning scheme starts with random trials, the information obtined from the trials should be used to direct the search towards regions in parameter space where solutions are more likely to be found. Unsupervised learning methods often depend on characterizing the performance of a controller by an *evaluation* function. The learning procedure modifies the controller to maximize the evaluation function using some kind of *hill climbing* method. Hill climbing can only be successful if the evaluation function is monotonously increasing towards better controllers. For speedy convergence it also has to have an appreciable gradient everywhere, otherwise training may get stuck or degenerate into a random search. We chose to investigate the behavior of time to failure for the linear controller in weight space. The results are shown in Figs. 2-5. In each figure the time to failure, from the same initial condition, is shown along 5 randomly oriented major circles (circles in planes that pass through the center) on the unit sphere in weight space. All circular trajectories in weight space start at the controller weights $w_\theta = 0.94$, $w_\omega = 0.33$, $w_x = 0.05$ and $w_v = 0.1$. Two initial conditions are shown for each of the two control regimes. One initial condition is *easy* and the other is *difficult*. Two things are apparent from these results. First, they confirm that for the easy initial conditions the controllers that do not fail occupy a relatively large area on the surface of the weight unit hypersphere. Second, they show that the time to failure when used as a performance evaluation function for hill climbing will cause problems. The time to failure has the form of a sharp peak on an otherwise flat area, and more importantly, there are local maxima. Particularly for the bang-bang regime (Fig. 2) where the gradually rising shoulders of the peak have what could be a chaotic structure.

With the difficult initial conditions a hill climbing method in the bang-bang regime (Fig. 3) will degenerate into a search for a very narrow peak in a flat space. For the continuous force the spines are certainly somewhat thicker but there are still many narrow peaks scattered through the flat region (Fig. 5).

Quality of Controllers

The time to failure on its own is not sufficient to characterize a good controller. The controllers that do not fail are of different quality. Some controllers do not center the cart or produce residual oscillations in both the angle and the position. The quality of a controller is determined by the time it needs to balance the pole and center the cart. Even among the controllers that finally balance the pole and center the cart some do it faster than others and with fewer oscillations. The better controllers exhibit *critical* damping in both θ and x. All these aspects are revealed by a graph of cart position and pole angle versus time. Fig. 7 shows such a graph for a continuous force controller capable of recovering from a difficult initial condition, barely avoiding hitting the end of the track. Evidently this controller delivers excellent control. It centers the cart in less than 8 s with

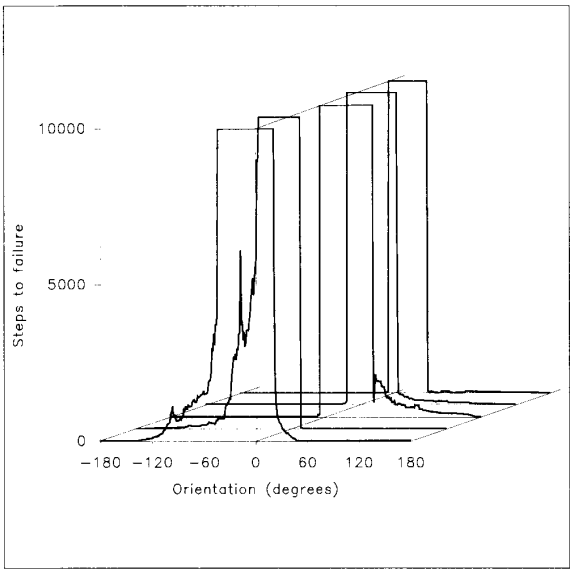

Fig. 2. Time to failure along a circular trajectory in weight space for bang-bang control. Easy initial condition: $x = 0$, $v = 0$, $\theta = 0.0$, $\omega = 0.1$. Starting point is our best unitary weight vector.

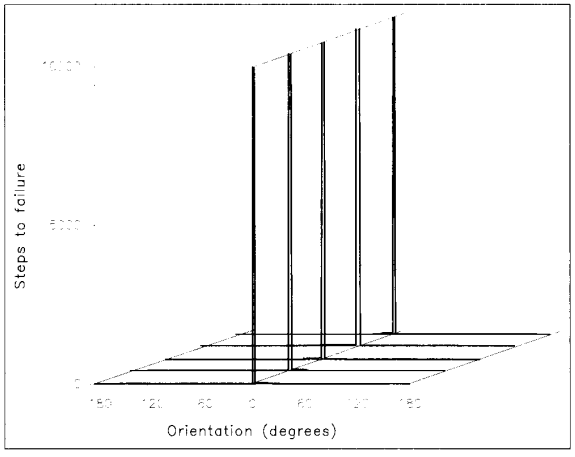

Fig. 3. Time to failure along a circular trajectory in weight space for bang-bang control. Difficult initial condition: $x = 1.0$, $v = 1.0$, $\theta = 0.1$, $\omega = 0.2$. Starting point is the best unitary weight vector.

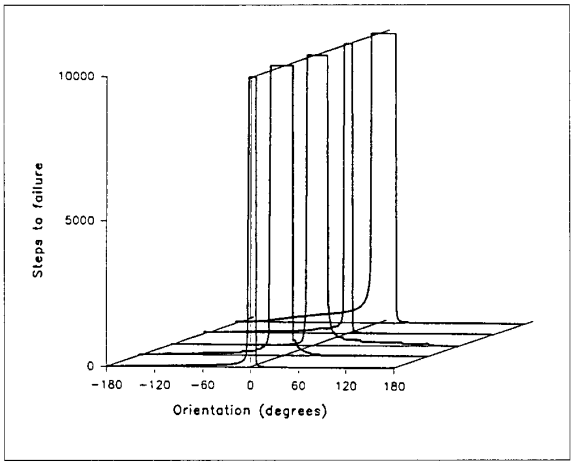

Fig. 4. Time to failure along a circular trajectory in weight space for continuous force control. Easy initial condition: $x = 0$, $v = 0$, $\theta = 0$, $\omega = 0.1$. Starting point is the best unitary weight vector.

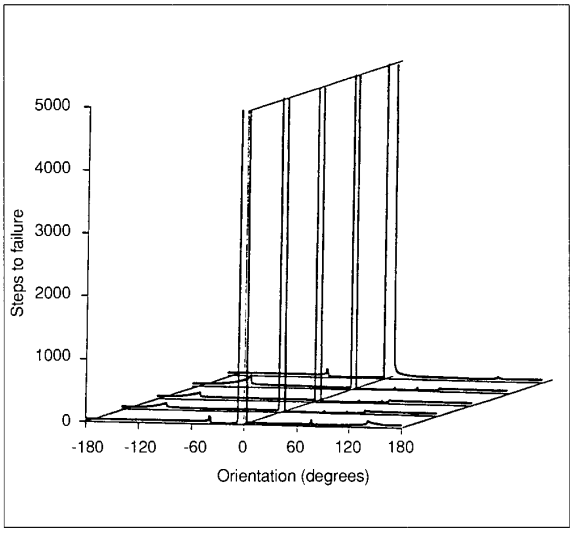

Fig. 5. Time to failure along a circular trajectory in weight space for continuous force control. Difficult initial condition: $x = 1.0$, $v = 1.0$, $\theta = 0.1$, $\omega = 0.2$. Starting point is the best unitary weight vector.

no overshooting and with no residual oscillation in neither position or angle. To describe the long term behavior of the controller we calculate the root mean square of the angle and the position over the first 1000 s. Fig. 8 shows that the controller still centers the cart when the force update interval is increased to 0.3 s, but there is a large residual oscillation as one would expect. Notice also that if the track ended at 2.4 m it would have failed in spite of being a reasonable controller. We removed the bounds of the track for these tests to prevent the cartpole from artificially failing by hitting the end of the track. The quality of the same controller in the bang-bang regime, shown in Fig. 9, is not nearly as good because it fails to center the cart. The RMS value for the position over the first 1000 s was 0.21 m. The controller also is less robust in the bang-bang regime. Fig. 10 shows that with a force update interval of 0.12 s it is no better than with 0.3 s in the continuous force regime.

One way to test the robustness of the controllers is to increase the interval between control force updates. Fig. 6 shows how the best controllers from Table II resist increasing update intervals. The superiority of the continuous force control is evident. The most resilient controllers are the same for both the bang-bang and the continuous force regime. The best controller was $w_\theta = 0.94$, $w_\omega = 0.33$, $w_x = 0.02$ and $w_v = 0.06$. We found that this controller could still be improved by slightly increasing the position and

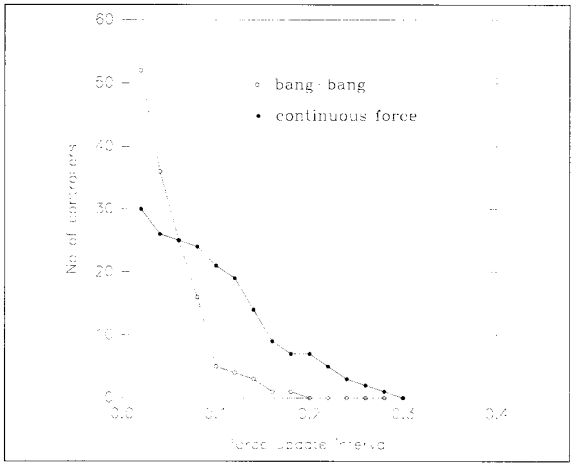

Fig. 6. Number of controllers, from the sample of 10 000 random unitary weight vectors with positive components, that prevent the cartpole from failing within 5 min. for increasing force update intervals. Initial condition: x = 1 m, v = 1 m/s, θ = 0.17 rad and ω = 0.1 rad/s.

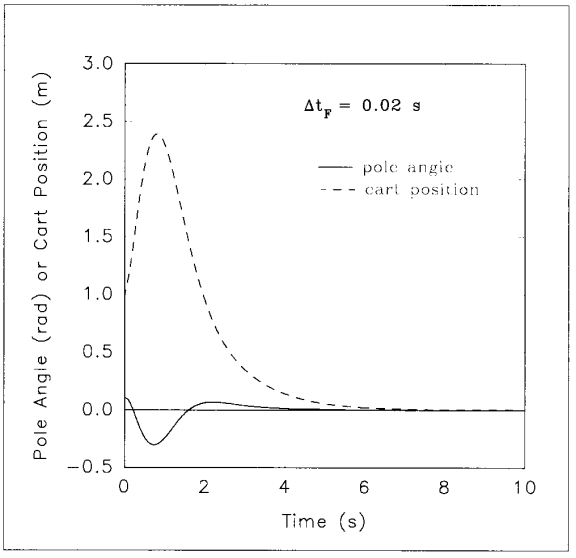

Fig. 7. Cart position and pole angle versus time for the best controller in the continuous force regime. Force update interval Δt_F = 0.02 s. Initial condition: x = 1 m, v = 1 m/s, θ = 0.1 rad and ω = 0.2 rad/s. RMS values over first 1000 s: $\bar{x}$ = 0.0076 m, $\bar{\theta}$ = 0.0768 rad.

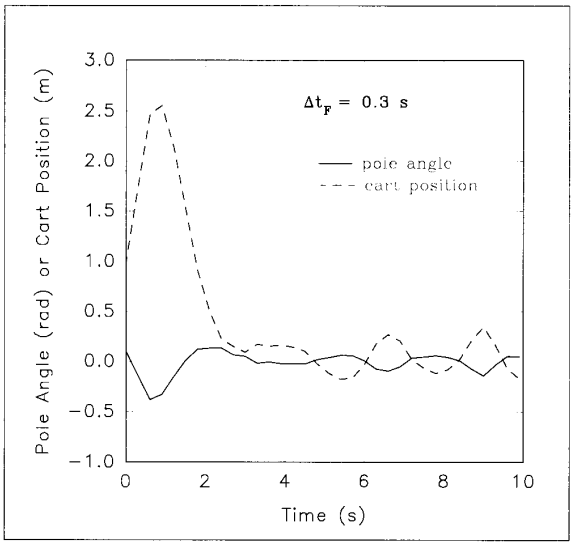

Fig. 8. Cart position and pole angle versus time for the best controller in the continuous force regime. Force update interval Δt_F = 0.3 s. Same initial conditions as in Fig. 7.

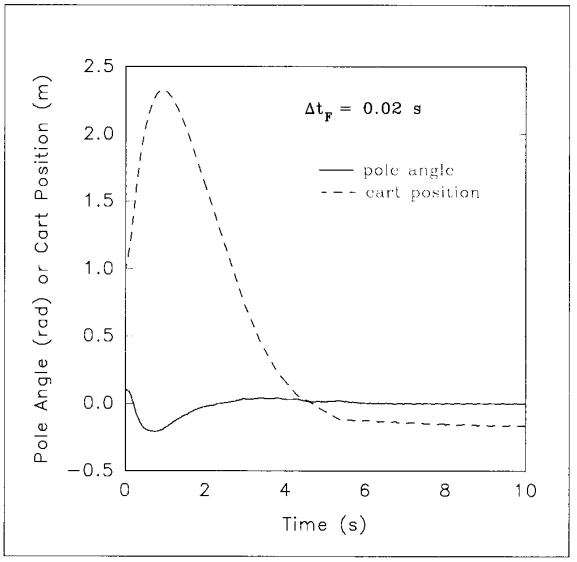

Fig. 9. Cart position and pole angle versus time for the best controller in the bang-bang regime. Force update interval Δt_F = 0.02 s. Same initial conditions as in Fig. 7. RMS values over first 1000 s: $\bar{x}$ = 0.21 m, $\bar{\theta}$ = 0.0061 rad.

velocity weights to $w_{\dot{x}} = 0.05$ and $w_{\dot{\theta}} = 0.1$. It then became our *best* controller.

The rapid deterioration of performance with increased update interval is simple to explain. With longer intervals the constant control force is applied for a longer time allowing the system to evolve to a state for which the control action is no longer correct. This is particularly severe for the bang-bang regime if the pole is close to the upright position. The full force is applied too long, pushing the pole far beyond the equilibrium position, possibly to a state where recovery is impossible even if all further control actions were correct. This happens, for example, when the force is too small to erect the pole.

The range of initial conditions from which the cartpole can recover is a further measure of robustness. The initial values of θ and ω are the most critical because a linear controller first brings the pole up and then centers it. Once the pole is close to

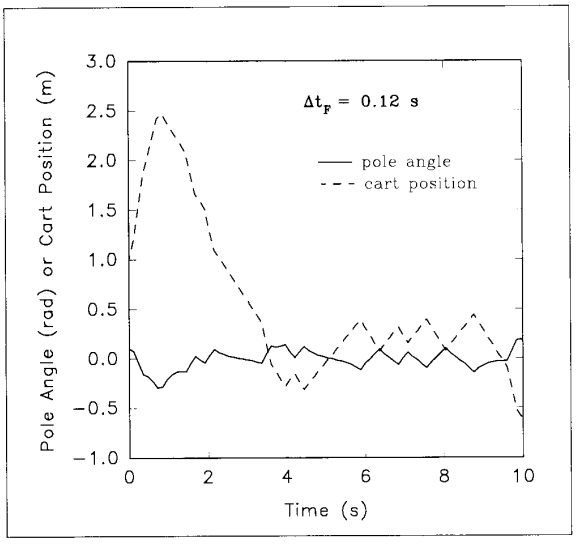

Fig. 10. Cart position and pole angle versus time for the best controller in the bang-bang regime. Force update interval $\Delta t_F = 0.12$ s. Same initial conditions as in Fig. 7.

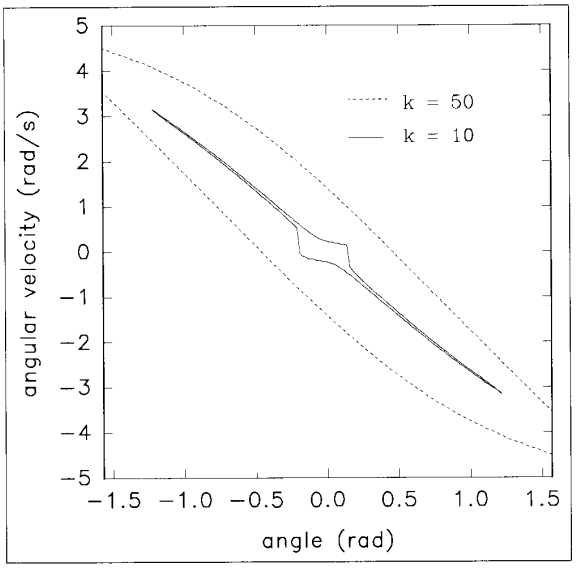

Fig. 11. Range of initial angular conditions for the pole from which our best linear controller can balance the pole and center the cart in the continuous force regime. The cart starts at rest at the center of the track. The region of initial conditions is delimited by the upper and lower curves for the respective values of k. The region (k = 50) is almost identical to the one obtained for bang-bang control.

upright it can always be centered unless the distance from the center or the linear velocity are large. It should be pointed out that imposing bounds on the track for the cartpole is largely irrelevant. Imposing reasonable bounds has the sole effect of making the problem always solvable by a linear controller. The values of x and v can not reach the region where the linear controller fails. Moreover the behavior of the cartpole with a linear controller is such that it will never move very far away from the center anyway, unless it is forced to do so by initially releasing it at a large distance or a large initial velocity. Given that the pole's initial state is the most important part of the initial condition it is sufficient to determine the range of θ and ω from which it can be erected. Fig. 11 shows the region in the $\theta - \omega$ plane from which our best controller would not fail for 300 s with the cart released from rest at the center of the track. As in all our experiments the linear force was limited to a maximum of 10 N, the value used by most other researchers, and the constant k was 50. Both the maximum force deliverable by the motor and the proportionality constant will affect the results. The behavior for the small value $k = 10$ in Fig. 11 illustrates this. A small value of k makes the motor insensitive to deviations from the target state, whereas a large value makes the motor response very sensitive. With a limit on the force, as in our experiments, a large k generates essentially bang-bang control by applying maximum force except for very small deviations.

A detailed analysis of the effect of these parameters remains to be done. The point we want to make with this figure is that the controller can handle a range of initial conditions much wider than expected. For example it has been assumed by Widrow and Smith [1], and explicitly stated by Guez and Selinsky [17] that a linear controller is only adequate for that region in state space where the system is well described by the linearized dynamic equations. These results show that this is not so.

Finally the question of sensitivity of the linear controller to errors in the measurement of the state variables has to be addressed. The robustness of controllers to weight changes as evidenced by the area occupied by good controllers on the surface of the hypersphere can be translated into robustness to errors in the state variables. This is because of the nature of the linear control law, where changes in the state variables are indistinguishable from changes in the weights. Some researchers have chosen to use two successive pairs of values of x and θ as input variables for the neurocontroller. Replacing v and ω by their approximations $\Delta\theta/\Delta t$ and $\Delta x/\Delta t$ transforms the linear control law to suit these inputs [5]. Again, a small error in the state variables is equivalent to a small error in the weights, and not surprisingly, this formulation does not make the problem harder. More anecdotal evidence for the robustness of the noise and errors in the state variables comes from the very crude cartpole built in our laboratory from toy components. This machine has all sorts of errors: time delays in the A/D conversion of the potentiometer reading for the pole angle, computation time delays, asymmetry in the force delivered by the motor in the forward and reverse direction, drag by the umbilical cable that connects it to the computer, and crude position measurement compounded by slip of the wheels. Nevertheless a linear controller capable of balancing the cartpole for more than 10 min at a time was easily found by random search.

Benchmark Problem

We showed in the previous section that a random search in weight space will quickly find suitable weights for a linear controller that balances the pole and centers the cart. The less

restrictive condition of avoiding failure in the standard cartpole problem has been used as evidence for some of the most widely quoted unsupervised learning schemes. We suggest that in view of our result successful balancing, and even centering, of the cartpole is not a convincing proof for effective learning. Instead the controller obtained from any learning scheme has to conform to more stringent performance criteria that will markedly reduce its likelihood of being found by chance.

A benchmark specification has to prescribe minimal reporting requirements that will allow the control engineer to judge the usefulness of the method for a particular application, regardless of the motivation of the researcher. The computer simulated cartpole as a benchmark for unsupervised training methods should meet the following requirements:

Standardized Computer Simulations

• Frictionless equations of motion (1) and (2).

In a real cartpole friction is difficult to determine. The performance of a controller should not rely on parameters hard to control in practice.

• Standard cartpole parameters as in Table I, except for pole failure angles.

The values in the table for the length and mass of the pole, mass of the cart, maximum force, and force update interval were often used in earlier work. The values are reasonable for a laboratory cartpole experiment.

• Failure angles of 90°.

The failure angles of ±12° are arbitrary and too restrictive. The maximum force the motor can deliver sets the real limits to angular conditions from which the cartpole can recover. Once the pole exceeds the limit it falls over quickly anyway. We suggest that the track limits are kept to enable comparison with earlier work. They are not really necessary but ±2.4 m provides ample room for succesful controllers.

Performance Description of the Controller

To quantify the performance of the controller release the cart-pole from the following initial state: x = 1 m, v = 1 m/s, θ = 0.1 rad and ω = 0.2 rad/s, and provide:

1. A plot of the pole angle and cart position vs time for the controlled cartpole over the first 20s (or less if the goal is achieved faster).

In case that the training method does not guarantee a symmetric solution a plot for the negative of this initial condition also is required.

2. The root mean square amplitude of the angle of the pole measured over the first 1000 s sampled at 1 s intervals or smaller.

3. The root mean square amplitude of the position of the cart measured over 1000 s at 1 s intervals or smaller.

4. Total training time and its standard deviation, in seconds, required to achieve the performance described in items 1 to 3.

The total number of trials is not a sufficient measure because of possible large differences in the duration of individual trials.

5. Boundaries in the $\theta - \omega$ plane from which the controller can balance the pole, when released with the cart at rest at the center of the track.

6. Demonstrate the behavior of the controller for force update intervals larger than 0.02 s.

Items 2 and 3 will characterize the long term behavior of the system.

Training Methods

As we have shown, a good neural controller for the cartpole experiment only requires finding four weights. Once one understands the strategy to be followed for balancing the pole and centering the cart, values for the weights can be found by trial and error in a short time. Alternatively, one can use a random search, or try to derive a set of suitable weights analytically, for example by linearizing the equation of motion, as was done by Widrow & Smith [1]. However the strength of neural network methods lies in algorithms that find the proper weights in a systematic way. These algorithms fall into two classes: supervised learning and unsupervised learning. Supervised learning methods need a teacher who knows the correct control action for every state of the system to be controlled (the teacher can be a human or another controller). The goal is to train the neural controller to imitate the teacher in a finite number of learning steps.

Unsupervised learning, on the other hand assumes that there are no examples of control actions that produce the desired behavior. The desired behavior of the system is only known as a goal and the correct control actions can only be inferred indirectly from experimenting with the system. Single input output pairs cannot be rated as wrong or correct, instead *sequences of actions* fail or succeed in achieving the goal. As the amount of information that is available to a learning system can vary, unsupervised learning methods are often divided into reinforcement learning and self-organization. In reinforcement learning the teacher is replaced by a critic or evaluation function that rates the actions taken by the learning system and rewards or punishes the system accordingly. Such an identifiable reward signal is absent in self-organization. A self-organizing system changes its response during its interaction with the environment according to some inbuilt learning dynamics that will work towards the achievement of a goal. The distinction often is subtle, therefore we refer to all methods that do not have a teacher as unsupervised.

Supervised Learning

Supervised learning is a function approximation problem. From a finite number of examples given by the teacher, the controller has to infer a mapping from the state space onto the control actions. The main purpose of most of the work on supervised learning for the cartpole was to show that a neural network can indeed be trained to imitate a teacher. This kind of demonstration is no longer needed. Theorems proven in recent years guarantee the existence of a neural network that approximates a given, reasonably well behaved function on a bounded domain [22]-[24]. For the cartpole we know that a linear control law is adequate. Therefore a single neurone is enough for a satisfactory controller. Given a teacher, all that has to be done is to adjust the network weights with an error minimization method.

This is what Widrow and Smith [1] did as early as in 1963. In their experiments they quantized and coded the state variables in a 6-bit linearly independent code, translating into 24 binary inputs. They used *bang-bang* control, suitable to the +1 or -1 output of the McCulloch-Pitts neurone. The output determined the direction a force of fixed magnitude exerted upon the cart. The neurone was trained with the δ-rule on a teacher signal. They

obtained the teacher signal by linearizing the dynamics of the system and applying conventional control theory methods. The force delivered by the teacher was of the linear classifier type.[1]

In a more recent work, Tolat and Widrow [5], instead of quantizing the state variables, used rough (5×11) pixel images of the cartpole as inputs to an ADALINE neurone. To convey the time evolution information they provided the pixel images at two successive time steps as inputs. The neurone had 110 inputs and mastered the classification task well enough to keep the pole from falling. Tolat and Widrow [5], unfortunately, do not give the parameters used in their cartpole emulation. The teacher was again of the linear classifier type. Their data, however, show that the pixel images of the cartpole do not form a totally linearly separable set, because even a least square fit with all the images produced a classification error of 1.78%.

The work where Widrow and Smith [1] show that a linear control law is sufficient to solve the problem, is well known. Nevertheless several researchers have used multilayer networks for approximating the control law. The justification for such work could be the expectation that a multilayer network could learn a better, nonlinear control law. However, there also seems to be a widely held misconception that the linear controller is only adequate in the narrow range where the linearized dynamic equations are a good approximation. Guez and Selinsky [17] trained a network with two hidden layers, 16 neurones in the first hidden layer and 4 neurones in the second hidden layer. They used several *teachers*: a linear control law derived from linearized dynamic equations, a nonlinear control law derived from the application of a feedback linearizing and decoupling transform, and finally a human teacher. Their results for the *linear* teacher seem as good as any other. They are definitely much better than those obtained with the human teacher. Unfortunately Guez and Selinsky [17] only compared the performance of the nonlinear and human teacher. Their results can not be directly compared with other work because they deviate from the Barto *et al.* [4] parameters by introducing a new dynamic friction term for the cart. Failure of giving the values for two parameters in their nonlinear control law also impedes the reproduction of their results.

Troudet and Merill [29] tested the ability of a multilayer Perceptron to extract the control law from noisy data, when the control law was deliberately transformed in such a way as to make it nonlinear. As could be expected the network learned a slightly different function than that provided by the teacher. Coincidentally it produced a marginally better performance in the cartpole.

Unsupervised Learning

Contrary to the largely solved problem of supervised learning, evidence for the effectiveness of unsupervised learning for the cartpole is scarce. Several unsupervised learning methods for the cartpole have been proposed. Some of them are very elaborate and their authors often do not give enough details in their papers to compare the methods and assess their relative merits. As shown earlier in this article, success in balancing the pole after a large number of trials is not a sufficient measure of learning performance.

[1] It seems that the weight values for θ and $\dot{\theta}$ published by Widrow [3] for their teaching controller contain a sign error.

Adaptive Critic

Barto, Sutton and Anderson (1983) [4] extended the idea of learning with a critic, first used in [18] for training a McCulloch-Pitts neurone to play the game of blackjack. Barto *et al.* [4] followed Widrow and Smith [1] in using *bang-bang* control for the cartpole experiment. They also build on Michie and Chambers' 1968 [20] BOXES scheme — the first heuristic attempt at unsupervised learning for the cartpole. An adaptive critic element (ACE) provides an evaluation of every state of the cartpole system that is used to steer the learning process of the controller. They attempt to solve the *credit assignment* problem by training the adaptive critic element (ACE) to predict failure, given the current state of the cartpole. There is no need for any specific knowledge about the dynamics of the cartpole system. Weights in both the ACE and the controller are adjusted in proportion to the *change* in prediction from one time step to the next. The failure signal only occurs at the end of a balancing trial. Initially all the weights of the ACE are set to zero, and consequently the prediction is zero for all states. Gradually, as more trials are done, nonzero predictions spread out from the final failure states. The controller is nondeterministic, its output biases a random process towards one of the two control actions. Barto *et al.* [4] followed the BOXES schem in quantising state space. Because the cartpole state is coded in a sparse vector of 162 component, one component for each of the 162 quantized regions of state space, a large number of weights have to be found. However, sparse coding makes weight adjustments mutually independent.

The training examples presented in Barto *et al.* [4] show that typically around 60 trial runs are needed before the controller is able to prevent the cartpole from failing. They compare the adaptive critic with the BOXES scheme under the same conditions. Their paper contains no evidence for the cart being properly centered by either method. The original Michie and Chambers paper privides no performance measure other than a weighted average of survival time versus accumulated learning time. The time to failure remains small for most of the learning trials, until it starts to increase steeply to a large value. There is no sign of gradual improvement of behavior. Furthermore, the learning scheme does not perform any better, or even differently, than a random search of weights for a linear controller. An argument against the hypothesis of a hidden random search could be that the likelihood of finding 162 weights would be very low. This is not convincing because during typical runs of the adaptive critic, as well as in the BOXES method, only a small number of state space cells are visited. Furthermore, the solution is not unique.

In 1989, Anderson [2] reported new experiments on cartpole balancing using an adaptive critic, but this time using the real valued state variables directly as inputs. The adaptive critic and the controller were two-layer networks trained with a variant of backpropagation. The results were far from impressive. The system needed 6000 (!) trials before the time to failure showed consistent increases. From then on time to failure increased steeply in a form similar to the results of Barto *et al.* [4]. Not surprisingly the output of the trained controller (action network) reported by Anderson [2] shows clearly that it behaves like a linear controller.

More researchers experimented in recent years with the basic ideas of the adaptive critic. Jameson 1990 [6] maximized the critic output by directly backpropagating the error over one time step to the controller via a neural network emulator of the cartpole

dynamics. Using a neural network to model a plant and then backpropagating the error to the controller is a technique that had been used in [7] for the *truck backer upper* problem. The emulator was trained to output the state change produced by a control action rather than the new state. In this way Jameson [6] needed 10 times fewer trials than Anderson [2] but still 10 times more than Barto *et al.* [4]. A controller was deemed successful if it would not fail within 2000 s. In addition, Jameson's [6] controller, as opposed to Anderson's [2], failed sporadically in trials where the cartpole was released close to the center of the track with the pole inclined at less than 12°. Jameson [6] could not eliminate these failures by further training.

Jordan and Jacobs 1990 [15] tested another variant of the adaptive critic. They model the critic by a multilayer neural network so that the error of the critic can be backpropagated to the controller. The virtues of their work are that they used the state variables directly as inputs and let the control force take continuous values. By using continuous force they could no longer start the balancing trials from a balanced and centered position as was done in earlier work. They used randomly selected initial conditions for each balancing trial, although they do not report the range of the initial conditions They report 8 learning trials, two of them failed to train, for the remaining 6 they give the learning curve (average time to failure versus trials). To train they needed from around 4000 to almost 30 000 trials. Although one would expect that the resulting controllers would not only balance the pole but also center the cart, Jordan and Jacobs [15] do not report on the quality of the controllers.

In a recent work, Lin and Kim [12] [13] used the CMAC technique for implementing the controller and the ACE. By doing so they can train the controller with fewer trials than Barto *et al.* [4], also their results show evidence of gradual improvement over trials. Unfortunately, Lin and Kim [12], [13] do not give details of the mapping of cartpole states to CMAC memory cells, leaving open the question of what is responsible for the better results. Is it due to better quantization of the input space, or what else?

Another apparently successful modification of Barto *et al.*'s [4] adaptive critic was reported in 1988 by Rosen, Goodwin, and Vidal [28]. They dispense with the critic element and instead add a new term to the weight adaptation equation that increases the weight of the boxes that are visited more frequently. This is based on the knowledge that when the cartpole is balanced it will move in a confined region of state space giving states, if they are quantized, a high chance to recur. The authors claim a 400% improvement in performance over the results by Barto *et al.* [4]. Again, few details of this work are available since only an abstract has been published.

The work published on the adaptive critic method seems to suggest that with further work a useful and consistent method could be developed. To assess the value of this and other learning control methods it is necessary to quantify the performance of the controllers in a comparable manner.

Explicit Evaluation Functions

Some researchers have tested evaluation functions specified on the basis of some knowledge of the behavior of the system. This approach lies somewhere between supervised and unsupervised learning. The weights of the controller are changed in the direction that increases the evaluation function.

Ritter, Martinetz and Schulten [26] use the evaluation function $R(\theta) = -\theta^2$ in conjunction with Kohonen's self-organizing feature map. Unfortunately they limit themselves to balancing the pole, irrespective of the final position of the cart. The feature map is an improvement over the simple BOXES vector quantization scheme introduced by Michie and Chambers [20]. The feature map vector quantization of input space gives higher resolution in the regions that are most critical. Instead of using the pole state variables θ and ω for input, they use the inclination of the pole at two successive time steps, $\theta(t)$ and $\theta(t + 1)$. Ritter *et al.* [26] associate to each cell or box in input space a force of variable magnitude. Whenever the pole visits a cell a small random noise is added to the force associated with that cell. Each cell keeps track of the average increment ΔR it has achieved so far. Only if the modified force results in a larger ΔR at the end of the time step adaptation of the codebook vector, the force and the amplitude of the random noise are adapted following Kohonen's adaptation rule. With 625 codebook vectors after 3000 training steps the controller can erect the pole from an angle of 40° quite well. The size of the time step for these experiments was an unusually large value of 0.3 s, which indicates very good control. The basic idea in Ritter *et al.*'s [26] work is essentially the same as that of Mitchie and Chambers. The differences are in the evaluation rule, the input quantization, and consistent use of Kohonen's adaptation rule.

Cotter *et al.* [25] attempted to minimize the objective function $\vec{U}(w) = 1/(1 + t)$, where t is the time to failure, through random searches and simulated annealing. Random searches always advanced from the best result attained so far. Random searches worked somewhat better than simulated annealing. The disappointing results they obtained, (the best controller maintained the pole upright for only 6.4 s!), are probably due to the unduly complex network of 20 fully interconnected neurones.

Genetic Algorithms

Genetic algorithms allow directed searches of weight space. Wieland [16] used genetic algorithms to generate controllers for the standard cartpole and some complications of it, like two poles on the same cart, and an articulated pole. For the standard cartpole he used a six-neurone fully interconnected network. For input he uses positions at two successive time steps. After six generations his population of controllers contained controllers capable of maintaining the pole within the limits set by the failure angles and preventing the cart from hitting the end of the track almost indefinitely (at least 5.5 h). Since no figures for the size of the initial population were given it is difficult to assess if the genetic search is more effective than a random search and if so by how much.

Maricic [14] used a genetic algorithm to find the weights for a neurocontroller with five fully interconnected neurones. Four neurones acted as input neurones receiving one of the normalised state variables, the sign of the output of the fifth neurone determined the direction of the constant magnitude control force. The evaluation function was the time the cartpole remained within 1m from the origin without falling over. Maricic [14] evaluated a total of 630 controllers spanning 21 generations. The maximum duration of each simulation was limited to 30 s. The best controller was capable of reaching the 30-s time limit from a broad range of initial conditions. The average time to failure for the geneti-

cally evolved controllers showed a gradual increase as more genetically evolved controllers were evaluated, reaching an average of 9 s over the 630 controllers. In comparison, the average time of the same number of randomly chosen controllers increased only almost imperceptibly.

As in Weiland [16], the results produced in Maricic [14] are inconclusive. It seems that the genetic search of weight space consistently produces a population of controllers that perform better on average than the controllers selected at random. However, the form Maricic [14] chose to present his results does not rule out that among the randomly selected controllers there might have been a good one. One good controller is all that is needed. The other question of course is why use a controller with 25 weights when one with 4 could do. It is clear that a random search for a controller with 25 weights will be much less likely to succeed than a search for one with 4 weights.

Dominic *et al.* [27] used a genetic algorithm to find the weights for a neural net controller with the same structure as the one used by Anderson [2] in his adaptive critic experiments. They found that a good controller could be obtained with the genetic algorithm in roughly the same number of trials (6000) needed in [2]. However the *genetic* controller consistently succeeded in around 10% more trials, from their chosen test sets of starting conditions, than the *adaptive critic* controller. For the comparison, failure was defined as a pole angle greater than 12° from the vertical. Beyond a failure angle of 35° the adaptive critic would no longer learn, whereas genetic controllers could still be found when the failure angle was set to 74°.

Genetic algorithms seem to succeed in the end even for networks that are more complex than necessary, but they do so at considerable computational expense. None of the authors describe how their genetically evolved controllers perform on centering.

Linear Controllers Easy to Find

We show that, for the cartpole experiment, it is easy to find, by simple random search in weight space, linear controllers that not only balance the pole but also center the cart. This result indicates that controlling the cartpole is not the difficult nonlinear problem assumed by many authors. Particularly, balancing without centering is a very forgiving control objective. The controllers are successful over a far larger range of initial conditions than had been assumed previously, nevertheless the quality of the control varies widely. In the case of continuously variable force, excellent controllers exist that balance and center in short time, with negligible residual oscillations. Such controllers are much harder to find by random search.

Many of the unsupervised learning schemes described in the literature require a training effort equal or larger than the random search reported here; measured in learning trials or total learning time. Unfortunately, data on the quality of the controllers obtained by unsupervised learning is very scarce. Therefore, there is little factual evidence that the unsupervised learning schemes perform better than the simple random search. Quantitative comparison between the various suggested methods of control seems to be a necessary step in research into learning controllers. To this end, we propose a modified version of the cartpole experiment as a benchmark problem for learning control, together with a set of reporting requirements. The specifications make it unlikely to find a satisfactory solution by random search.

References

[1] B. Widrow and F.W. Smith, "Pattern recognizing control systems," in *1963 Computer Info. Sci. (COINS) Symp. Proc.*, Washington, DC, 1963, pp. 288–317.

[2] C.W. Anderson, "Learning to control an inverted pendulum using neural networks," *IEEE Control Syst. Mag.*, vol. 15, pp. 31-36, Apr. 1989. [3] B. Widrow, "The original adaptive neural net broom-balancer," in *Proc. IEEE Int. Symp. Circuits Syst.*, Philadelphia, PA, May 1987, pp. 351-357.

[4] A.G. Barto, R.S. Sutton, and C.W. Anderson, "Neuronelike adaptive elements that can solve difficult learning control problems," *IEEE Trans. Syst., Man., Cybern.*, vol. SMC-13, pp. 834–846, 1983.

[5] V.V. Tolat and B. Widrow, "An adaptive 'broom balancer' with visual inputs," in *Proc. Int. Conf. Neural Networks,* San Diego, CA, July 1988, pp. II-641–II-647.

[6] J. Jameson, "A neurocontroller based on model feedback and the adaptive heuristic critic," in *Proc. Int. Joint Neural Network Conf. (IJCNN)*, San Diego, CA, June 1990, pp. II-359–II-364.

[7] D.H. Nguyen and B. Widrow, "Neural networks for self-learning control systems," *IEEE Control Syst. Mag.*, pp. 18-23, Apr. 1990.

[8] C.C. Lee and H.R. Berenji, "An intelligent controller based on approximate reasoning and reinforcement learning," in *Proc. IEEE Int. Symp. Intell. Control*, 1989, pp. 200–205.

[9] R.H. Cannon, Jr., *Dynamics of Physical Systems.* New York: McGraw-Hill, 1967.

[10] G.J. Wang and D.K. Miu, "Unsupervising adaptation neural-network control," in *Proc. Int. Joint Neural Network Conf. (IJCNN)*, San Diego, CA, June 1990, pp. III-421–III-428.

[11] A. Patrikar and J. Provence, *"A self-organizing controller for dynamic processes using neural networks,"* in *Proc. Int. Joint Neural Network Conference (IJCNN)*, San Diego, CA, June 1990, pp. III-359–III-364.

[12] C.S. Lin and H. Kim, *"Use of CMAC neural networks in reinforcement self-learning control,"* in *Proc. Conf. Artificial Neural Networks*, Kohonen *et al.*, Eds. Elsevier, 1991, pp. 1285–1288.

[13] C.S. Lin and H. Kim, "CMAC-based adaptive critic self-learning control," *IEEE Trans. Neural Networks*, vol. 2, pp. 530-533, 1991.

[14] B. Maricic, "Genetically programmed neural network for solving pole-balancing problem," in *Proc. Artificial Neural Networks*, Kohonen *et al.*, Eds. Elsevier, 1991, pp. 1273–1276.

[15] M.I. Jordan and R.A. Jacobs, "Learning to control an unstable system with forward modeling" in *Advances in Neural Information Processing Systems 2*, D.S. Touretzky, Ed. Morgan Kaufmann, 1990, pp. 324–331.

[16] A.P. Wieland, "Evolving neural network controllers for unstable systems," in *Proc. Int. Joint Conf. Neural Networks (IJCNN)*, Seattle, WA, 1991, pp. II-667–II-673.

[17] A. Guez and J. Selinsky, "A trainable neuromorphic controller," *J. Rob. Syst.*, vol. 5, 1988, pp. 363–388.

[18] B. Widrow, N.K. Gupta, and S. Maitra, "Punish/reward: Learning with a critic in adaptive threshold systems," *IEEE Trans. Syst., Man, Cybern.*, vol. 3, 1973, pp. 455–465.

[19] R. Hecht-Nielsen, *Neurocomputing.* Reading, MA: Addison-Wesley, 1990, p. 343.

[20] D. Michie and R.A. Chambers, "BOXES: An experiment in adaptive control," in *Machine Intelligence 2*, E. Dale and D. Michie, Eds. Edinburgh, U.K.: Oliver and Boyd, 1968, pp. 137–152.

[21] C.W. Anderson and W.M. Miller, "Challenging control problems," in *Neural Networks for Control*. Miller, Sutton and Werbos, Eds. Cambridge, MA: M.I.T. Press, 1990.

[22] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, pp. 359–366, 1989.

[23] K. Funahashi, "On the approximate realization of continuous mapping by neural networks," *Neural Networks*, vol. 2, pp. 183–192, 1989.

[24] G. Cybenko, "Approximation by superposition of a sigmoidal function," *Math. Control, Signals Syst.*, vol. 2, pp. 303–314, 1989.

[25] N.E. Cotter, T.M. Guillerm, J.B. Soller, and P.R. Conwell, "Perjudicial searches and the pole balancer," *Proc. Int. Joint Neural Network Conf. (IJCNN)*, Seattle, WA, July 1991, pp. II-689–II-694.

[26] H. Ritter, T. Martinetz, and K. Schulten, *"Neuroneale Netze."* Addison-Wesley, 1990, pp. 115–131.

[27] S. Dominic, R. Das, D. Withley, and C. Anderson, "Genetic reinforcement learning for neural networks," in *Proc. Int. Joint Conf. Neural Networks*, Seattle, WA, July 1991, pp. II-71–II-76.

[28] B.E. Rosen, J.M. Goodwin, and J.J. Vidal, "State recurrence learning," presented at First Ann. Meet. Int. Neural Network Soc., Boston, MA, 1988.

[29] T. Troudet and W. Merill, "Neuromorphic learning of continuous-valued mappings from noise-corrupted data," in *IEEE Trans. Neural Networks*, vol. 2, pp. 294–301, 1991.

[30] I. Bratko, "Qualitative modelling: Learning and control," presented at 6th Czechoslovak Conf. Art. Intell., Prague, Czechoslovakia, June 1991.

Shlomo Geva was born in Israel in 1954. He obtained the B.Sc. degree in chemistry and physics from the Hebrew University in 1981, and the Master of Applied Science degree in computing from the Queensland University of Technology, Australia, in 1987. He is a Senior Lecturer in the School of Computing Science, Queensland University of Technology. His research interests are in neural networks, their application in control, and machine learning.

Joaquin Sitte received the Licenciado degree in physics from the Universidad Central de Venezuela in 1968 and the Ph.D. degree in quantum chemistry from Uppsala University, Sweden, in 1974. Until 1985 he was an Associate Professor at the Universidad de Los Andes, Merida, Venezuela, where he led the Surface Physics Research Group. Since 1986 he has been on the faculty of the School of Computing Science, Queensland University of Technology, Australia. His main research interests are self-organizing and autonomous systems, machine learning, and neural networks.