Author(s): Matieni, X. and Dodds, Stephen J.
Title: A comparison of fixed final time optimal control computational methods with a view to closed loop implementation using artificial neural networks
Year of publication: 2009
Citation: Matieni, X. and Dodds, S.J. (2009) ‘A comparison of fixed final time optimal control computational methods with a view to closed loop implementation using artificial neural networks’ Proceedings of Advances in Computing and Technology, (AC&T) The School of Computing and Technology 4th Annual Conference, University of East London, pp.151-159
Link to published version:
http://www.uel.ac.uk/act/proceedings/documents/FinalProceedings.pdf

A COMPARISON OF FIXED FINAL TIME OPTIMAL CONTROL COMPUTATIONAL METHODS WITH A VIEW TO CLOSED LOOP IMPLEMENTATION USING ARTIFICIAL NEURAL NETWORKS

X Matieni, S J Dodds
Control Research Group, SCoT.
xamat@hotmail.com; stephen.dodds@spacecon.co.uk; s.j.dodds@uel.ac.uk

Abstract: The purpose of this paper is to lay the foundations of a new generation of closed loop optimal control laws based on the plant state space model and implemented using artificial neural networks. The basis is the long established open loop methods of Bellman and Pontryagin, which compute optimal controls off line and apply them subsequently in real time. They are therefore open loop methods and during the period leading up to the present century, they have been abandoned by the mainstream control researchers due to a) the fundamental drawback of susceptibility to plant modelling errors and external disturbances and b) the lack of success in deriving closed loop versions in all but the simplest and often unrealistic cases. The recent energy crisis, however, has promoted the authors to re-visit the classical optimal control methods with a view to deriving new practicable closed loop optimal control laws that could save terawatts of electrical energy by replacement of classical controllers throughout industry. First Bellman's and Pontryagin's methods are compared regarding ease of computation. Then a new optimal state feedback controller is proposed based on the training of artificial neural networks with the computed optimal controls.

1. Introduction:

After the establishment of classical linear feedback control theory and practice in the 1940s, attention was turned to establishing systematic methods for adjusting the parameters of controllers to improve their performances, leading to the concept of minimising (or maximising) a performance criterion (Bellman, 1957). Towards the end of the 1950s and during the 1960s, the so called modern control theory evolved centred on the concept of the state of a dynamical system, which proved to be of fundamental importance as this formed the foundation stones of *optimal control theory*. One of the drawbacks of this theory, however, is that closed loop versions in the form of state feedback control laws offering robustness with respect to external disturbances and plant modelling uncertainties are not readily attainable. Such control laws, however, have been derived for the time optimal control of a limited range of plants (Ryan, 1982).

The structure of the general optimal control problem can be described as follows:

a) The plant or process to be controlled has the following state space model:

$$\begin{cases} \dot{\mathbf{x}}(t) = f\left[\mathbf{x}(t), \mathbf{u}(t), t\right] \\ \mathbf{y}(t) = h\left[\mathbf{x}(t), t\right] \end{cases} \tag{1}$$

where $\mathbf{x} \in \Re^n$ is the plant state, $\mathbf{u} \in \Re^r$ is the set of control inputs and $\mathbf{y} \in \Re^m$ is the set of measured outputs.

b) There are restrictions on $\mathbf{u}(t)$ and $\mathbf{x}(t)$.

c) A reference signal, $\mathbf{r}(t)$, is provided that $\mathbf{y}(t)$ is intended to follow.

d) A performance criterion, sometimes called a cost function, has to be minimised, having the following general form,

$$J = \int_{t_0}^{T} G\left[\mathbf{x}(t), \mathbf{u}(t), \mathbf{r}(t), t\right] dt. \qquad (2)$$

where T is the end time of the control action. In (2), the scalar integrand, $G(\bullet)$, is referred to as the loss function and is a measure of instantaneous change in the performance. Hence the performance criterion is sometimes referred to as the cumulative loss.

The optimal control problem consists of the determination of the control input, $\mathbf{u}(t) = \mathbf{u}^o(t)$ that minimises the performance criterion, *J*, subject to constraints on $\mathbf{u}(t)$ and $\mathbf{x}(t)$ imposed by the plant hardware.

Two fundamental approaches to solving the optimal control problem were originated in the USA (Bellman, 1957) and in Russia (Pontryagin, 1959) and these will be described in the following two sections as they are proposed by the authors to form the basis of new closed loop optimal control laws implemented using artificial neural networks.

2. Dynamic Programming:

A simple regulator problem will serve to illustrate the dynamic programming approach. A *regulator*, in contrast a *controller*, is required maintain the output, $\mathbf{y}(t)$, fixed. The reference input is therefore $\mathbf{r}(t) = \mathbf{const}$.

The following example will suffice to demonstrate the method:

a) A single input, single output plant is considered having first-order linear dynamics, with the output, *y*, as the state variable:

$$\dot{y}(t) = Ay(t) + Bu(t) \qquad (3)$$

b) Since deviations of *y* from zero are minimised the performance criterion is

$$J = \int_{0}^{\infty} y^2(t) dt \qquad (4)$$

.

c) the controlled input has practical saturation constraints:

$$-u_{\max} \le u(t) \le u_{\max} \qquad (5)$$

The scalar Hamilton-Jacobi equation has to be satisfied when *J* has the required minimum value, J^o (Bellman, 1957) and (Bellman, et. al, 1962):

$$\frac{\partial J^o}{\partial t_o} + \frac{\partial J^o}{\partial y} f\left(y(t), u^o(t)\right) + G\left(y(t), u^o(t), r(t), t_o\right) = 0 \qquad (6)$$

From the statement of the problem, the terms of (6) are:

$$\frac{\partial J^o}{\partial t_o} = 0, \qquad (7)$$

$$f\left(y(t), u^o(t)\right) = \dot{y}(t) = Ay(t) + Bu(t) \qquad (8)$$

$$G\left(y(t), u^o(t), r(t), t_o\right) = y^2(t) \qquad (9)$$

Substituting (7), (8) and (9) into (6) gives:

$$\frac{\partial J^o}{\partial y}\left[Ay(t) + Bu(t)\right] + y^2(t) = 0 \qquad (10)$$

This expression must be minimised with respect to *u(t),* subject to the constraint

$$-u_{\max} \le u(t) \le u_{\max} \qquad (11)$$

Thus

$$\min_{u(t)}\left[\frac{\partial J^o}{\partial y}\left(Ay(t) + Bu(t) + y^2(t)\right)\right] = \min_{u(t)}\left[\frac{\partial J^o}{\partial y} Bu(t)\right] \qquad (12)$$

Hence the optimal control must satisfy

$$u^o(t) = \begin{cases} +u_{\max} & for \frac{\partial J^o}{\partial y} B < 0 \\ -u_{\max}, & for \frac{\partial J^o}{\partial y} B > 0 \end{cases} \Rightarrow$$

$$u^o(t) = -u_{\max}\,\mathrm{sgn}\left(B\frac{\partial J^o}{\partial y}\right) \qquad (13)$$

Substituting (14) into (10) gives:

$$\frac{\partial J^o}{\partial y}\left[Ay(t) - B\,\mathrm{sgn}\left(B\frac{\partial J^o}{\partial y}\right)\right] + y^2(t) = 0 \quad (14)$$

Assuming $B > 0$, this simplifies to:

$$Ay(t)\frac{\partial J^o}{\partial y} - \left|B\frac{\partial J^o}{\partial y}\right| + y^2(t) = 0 \qquad (15)$$

This is a nonlinear partial differential equation that has to be solved for J^o. The solution is then substituted into (14) in order to obtain the optimum $u^o(t)$. Most importantly, however, Bellman concluded that this demands the aid of a digital computer except for simple cases.

3. Pontryagin's Maximum Principle:

To explain this method, let the general plant (1) be expressed in the following component form:

$$\dot{x}_i(t) = f_i\left[\mathbf{x}(t), u(t)\right],\ i = 1, 2\cdots, n \quad (16)$$

The performance index to be minimised is as in the previous sections:

$$J = \int_0^T G\left[\mathbf{x}(t), u(t), r(t), t\right]dt, \qquad (17)$$

The maximum principle requires that the optimal control input, $u^o(t)$, that minimises J will maximise the scalar Hamiltonian function

$$\begin{aligned} H(t) = &\sum_{i=1}^{n} p_i(t) f_i\left[\mathbf{x}(t), u(t)\right] \\ &- G\left[\mathbf{x}(t), u(t), r(t), t\right] \end{aligned} \qquad (18)$$

where $p_i(t)$ is the *co-state* and obeys the following set of differential equations:

$$\dot{p}_i(t) = \frac{\partial H(t)}{\partial x_i(t)}, i = 1, 2, \cdots, n \qquad (19)$$

System (19) is often called the *adjoint system* and its state $\mathbf{p} \in \Re^n$ is known as the *co-state.*

From (16) and (18), $\dot{y}_i(t)$ can also be expressed in terms of H and p_i, as follows:

$$\dot{x}_i(t) = \frac{\partial H(t)}{\partial p_i(t)} \quad i = 1, 2, \cdots, n \qquad (20)$$

The necessary conditions of the maximum principle can be obtained from the dynamic programming equations by a simple change of variables. Reconsidering the Hamiltonian-Jacobi equation (6):

$$p_i(t) = -\frac{\partial J^o}{\partial y_i} \qquad (21)$$

and

$$H(t) = \frac{\partial J^o}{\partial t_0} \qquad (22)$$

where $t = t_0$ is the initial time. Substituting (21) into (22) yields

$$\begin{aligned} H(t) = &\sum_{i=1}^{n} p_i(t) f_i\left[\mathbf{x}(t), u(t)\right] \\ &- G\left[\mathbf{x}(t), u(t), r(t), t_o\right] \end{aligned} \qquad (23)$$

which is the basic Hamiltonian function (18). In addition, (19) can be obtained from (21) and (22) as follows:

$$\begin{aligned} \dot{p}_i(t) &= \frac{dp_i}{dt_o} = \frac{d}{dt_o}\left(-\frac{\partial J^o}{\partial y_i}\right) \\ &= -\frac{\partial^2 J^o}{\partial t_o \partial y_i} = -\frac{\partial H(t)}{\partial y_i(t)}. \end{aligned} \qquad (24)$$

Hence the basic maximum-principle relationships defined by (18) and (19) have been justified from the dynamic programming relationship (i.e., the Hamiltonian-Jacobi equation) by a simple change of variables. It is important to note, however, that this is not a proof of the maximum principle because this derivation assumed the existence of the derivatives

$$\partial J^o \big/ \partial y_i \text{ and } \partial J^o \big/ \partial t_o .$$

On the contrary, there are many cases where they do not exist but there is a proof (Bolttyanskii et.al., 1960).

The application of the maximum principle will now be illustrated by means of the same example used in Section 2. Thus the plant is

$$\dot{y}(t) = Ay(t) + Bu(t) \tag{25}$$

with performance criterion

$$\mathrm{J} = \int_0^\infty \mathrm{y}^2(\mathrm{t})\,\mathrm{dt} \tag{26}$$

and control saturation constraints given by

$$-u_{\max} \le u(t) \le u_{\max} \tag{27}$$

From the above, the values of $f_i\left[\mathbf{x}(t), u(t)\right]$ and $G\left[y(t), u(t), r(t), t_o\right]$ in the Hamiltonian equation are given by

$$\mathrm{f_i}\left[\mathrm{y(t)}, \mathrm{u(t)}\right] = \mathrm{Ay(t)} + \mathrm{Bu(t)} \tag{28}$$

$$\mathrm{G}\left[\mathrm{y(t)}, \mathrm{u(t)}, \mathrm{r(t)}, \mathrm{t_o}\right] = \mathrm{y}^2(\mathrm{t}). \tag{29}$$

Substituting (28) and (29) into the Hamiltonian equation, gives

$$\mathrm{H(t)} = \mathrm{p(t)}\left[\mathrm{Ay(t)} + \mathrm{Bu(t)}\right] - \mathrm{y}^2(\mathrm{t}) \tag{30}$$

Applying the fundamental relationship of the maximum principle

$$\dot{p}_i(t) = -\frac{\partial H(t)}{\partial x_i(t)} \tag{31}$$

to (30) yields the first equation to be solved:

$$\dot{p}(t) = -\frac{\partial H(t)}{\partial y(t)} = -p(t)A + 2y(t) \tag{32}$$

The second equation can be obtained from the fact that the Hamiltonian is maximised when the performance criterion is minimised. Maximising (30) with respect to *u(t)*:

$$\max_{u(t)} \left[p(t) Bu(t).\right] \tag{33}$$

and hence

$$u^o(t) = \operatorname{sgn}(Bp(t)) \tag{34}$$

This is the equivalent value for $u^o(t)$ in (13) obtained using dynamic programming. Substituting (21) into (34) gives the second necessary condition:

$$\dot{\mathrm{y}}(\mathrm{t}) = \mathrm{Ay(t)} + \mathrm{B}\operatorname{sgn}\left[\mathrm{Bp(t)}\right]. \tag{35}$$

The solution to this optimal control problem by the maximum principle has been reduced to the solution of the nonlinear ordinary differential equations (32) and (35) for *p(t),* which would then be substituted into (34) in order to obtain the optimum $u^o(t)$.

It is important to note, however, that the solution of the optimal control problem using the maximum principle entails finding the initial co-state, $\mathbf{p}(t_0)$, and in all but a few simple cases, a digital computer is required to produce a numerical solution iteratively. This conclusion is similar to that stated at the end of Section 2 for dynamic programming.

4. Minimum-Energy Problem:

Some work has been done on optimal control calculations aimed at minimising the control energy, exemplified by Pearson (1962). Since this era, however, interest in optimal control has reduced in view of the practical implementation difficulties discussed in the following section but in view of the current importance of energy conservation, this section considers a simple mechanism whose position is to be controlled, typical of those in industry. The state space model is as follows:

$$\left\{ \dot{x}_1 = x_2,\ \dot{x}_2 = \frac{1}{M}\left(K_a u - F x_2\right) \right. \qquad (36)$$

where x_1 is the position to be controlled, u is the control voltage, K_a is the actuator force constant, M is the mass of the moved object and F is the coefficient of viscous friction. The usual control saturation constraints (5) are assumed to apply.

It is commonly quoted that the square of the control signal utilised is proportional to power, and the time integral of power is energy. For minimising the energy, therefore, the performance criterion is usually taken as

$$J = \int_0^T u^2(t)dt \qquad (37)$$

Here the loss function represents power

$$\mathrm{G}\left[\mathbf{x}(\mathrm{t}),\mathrm{u}(\mathrm{t}),\mathrm{r}(\mathrm{t}),\mathrm{t}\right] = \mathrm{u}^2(\mathrm{t}) \qquad (38)$$

In this case, the Hamiltonian function is:

$$\begin{aligned} H(t) &= p_1(t)x_2(t) \\ &+ p_2(t)\left[K_a u(t) - F x_2(t)\right]/M - u^2(t) \end{aligned} \qquad (39)$$

The co-state equations are given by applying (19) to (39):

$$\dot{p}_1 = 0,\ \dot{p}_2 = p_1 - (F/M)p_2 \qquad (40)$$

The general solution may therefore be found analytically:

$$p_2(t) = p_1(0)\left(1 - e^{-Ft}\right)\frac{M}{F} + p_2(0)e^{-Ft} \qquad (41)$$

Since the terms $p_1(t)x_2(t)$ and $(F/M)p_2(t)x_2(t)$ in (39) are independent of the input *u(t),* the maximisation of the Hamiltonian function is given by

$$\max_{u(t)}\left[\left(K_a/M\right)p_2(t)u(t) - u^2(t)\right] \qquad (42)$$

From (40), there are two cases in which the Hamiltonian function can be maximised: Suppose $|u| < u_{\max}$. Then to find the optimal value, $u^o(t)$, of $u(t)$, (41) is differentiated:

$$\frac{\partial}{\partial u}\left[\left(K_a/M\right)p_2 u(t) - u^2(t)\right] = 0 \Rightarrow$$

$$u^0(t) = \frac{K_a}{2M}p_2(t) \qquad (43)$$

Therefore, if

$$\left|p_2(t)\right| \le 2\frac{M}{K_a}u_{\max},\ u^0(t) = \frac{K_a}{2M}p_2(t)$$

On the other hand, if $\left|p_2(t)\right| > 2\frac{M}{K_a}u_{\max}$, then

$$u^o(t) = u_{\max}\operatorname{sgn}\left[p_2(t)\right] \qquad (44)$$

These results indicate that the control is continuous over a fixed range and saturates whenever $\left|p_2(t)\right| > 2\frac{M}{K_a}u_{\max}$.

For cost function (37), the solution has been obtained analytically but for a given initial state, $\mathbf{x}(0)$, the corresponding initial co-state, $\mathbf{p}(0)$, is needed to evaluate (41) and hence determine $u^o(t)$ via (43) and (44).

At this point, however, *the authors wish to draw attention to the impracticality of the use of cost function* (37) *for minimum energy optimal control*: If the drive of the motion control system is regenerative, i.e., when the mechanism is slowed down, the excess kinetic energy is returned to the electric power supply, then the only energy to be minimised is that dissipated as heat due to the viscous friction and this should be taken as the loss function, which is $Fx_2^2(t)$. The cost function to be minimised is then the true energy loss:

$$J = \int_0^T F x_2^2(t)\,dt \qquad (45)$$

The corresponding Hamiltonian function is:

$$\begin{aligned} H(t) &= p_1(t)x_2(t) \\ &+ p_2(t)\left[K_a u(t) - F x_2(t)\right]/M - F x_2^2(t) \end{aligned} \qquad (46)$$

The co-state equations are

$$\dot{p}_1 = 0,\ \dot{p}_2 = p_1 - (F/M)\,p_2 - 2Fx_2 \qquad (47)$$

In contrast to (40), an analytical solution to (46) is not possible due to the presence of the plant state variable, x_2.

In view of (45), the optimal control is given by

$$\max_{u(t)}\left[\frac{K_a}{M} p_2(t)u(t)\right] \qquad (48)$$

Even for this simple example, the optimal control, $u^o(t)$, could only be determined numerically since no analytical solution exists for (46).

5. Towards a Practicable Optimal Control Strategy:

5.1 Open Loop and Closed Loop Control:

It is evident from the previous sections that Bellman's dynamic programming and Pontryagin's maximum principle are both yield optimal *but open loop* control. They predict $\mathbf{u}^o(t)$, and the corresponding state trajectory, $\mathbf{x}(t)$, of plant (1) given an accurate plant model and the initial plant state, $\mathbf{x}(0)$. So $\mathbf{u}^o(t)$ would be first computed and then applied in real time. This, however, would be impracticable due to plant modelling errors and an unknown disturbance, $\mathbf{d}(t)$, the true plant being represented by

$$\begin{cases} \dot{\mathbf{x}}'(t) = \tilde{f}\left[\mathbf{x}'(t), \mathbf{u}(t), \mathbf{d}(t), t\right] \\ \mathbf{y}'(t) = \tilde{h}\left[\mathbf{x}'(t), t\right] \end{cases} \qquad (49)$$

where $\tilde{f}[\bullet]$ and $\tilde{h}[\bullet]$ are, respectively, estimates of $f[\bullet]$ and $h[\bullet]$ in (1). Applying a control function, $\mathbf{u}^o(t)$, to plant (49) calculated using plant model (1) would result in the true state trajectory, $\mathbf{x}'(t)$, departing from the calculated trajectory, $\mathbf{x}(t)$. This departure would continue due to no information about the true plant behaviour being fed back to the control computer. Such *open loop control* is well known to be sensitive to plant modelling errors and disturbances. *Closed loop control* is highly desirable since it can partially compensate for such imperfections and prevent unbounded error build up between $\mathbf{x}'(t)$ and $\mathbf{x}(t)$. The reader might be tempted to try the simple approach of designing a model reference controller (MRC) acting on the known error between $\mathbf{y}'(t)$ and $\mathbf{y}(t)$, as shown in Fig. 1.

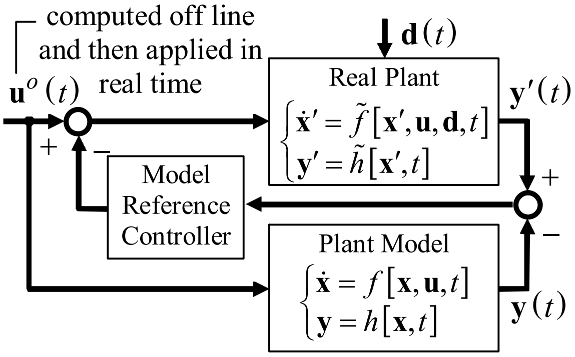

computed off line
and then applied in
real time
$\mathbf{d}(t)$
$\mathbf{u}^o(t)$
Real Plant
$\dot{\mathbf{x}}' = \tilde{f}[\mathbf{x}', \mathbf{u}, \mathbf{d}, t]$
$\mathbf{y}' = \tilde{h}[\mathbf{x}', t]$
$\mathbf{y}'(t)$
+
−
+
−
Model
Reference
Controller
Plant Model
$\dot{\mathbf{x}} = f[\mathbf{x}, \mathbf{u}, t]$
$\mathbf{y} = h[\mathbf{x}, t]$
$\mathbf{y}(t)$

Fig.1. Attempt to apply MRC to reduce and limit errors in open loop optimal control.

Although this may work in relatively simple cases, it is not recommended since the design of a model reference controller acting only on the error, $\mathbf{y}'(t) - \mathbf{y}(t)$, would, in general, be difficult for some nonlinear or relatively high order plants so this is not considered further. Instead the authors will proceed with the state feedback approach described below.

It is well known that *full state control* of a plant can yield the best possible control, because the plant state contains all the information about its instantaneous dynamic behaviour. Indeed, the optimal control calculations presented previously yield the future control, $\mathbf{u}^o(t)$, to apply, given the initial *state*, $\mathbf{x}(0)$, and a constant reference input, $\mathbf{r}$. If a closed form solution to the general optimal control problem were to exist, then the optimal control law would be a *state feedback control law*:

$$\mathbf{u}^o = \mathbf{G}^o(\mathbf{x},\mathbf{r}) \qquad (50)$$

It is important to note, however, that an observer would be needed to estimate all the state variables not directly measured. The following sub-section presents an approach to obtaining a closed loop state control law aimed at closely approximating the ideal one of (50). As stated previously, the loop closure affords some degree of robustness against plant modelling errors and external disturbances but how much depends upon the particular case and this will be determined in future work based on the method presented in the following section.

5.2 State Control based on Artificial Neural Networks:

Suppose a particular optimal control value, $\mathbf{u}^o$, has been calculated off line that should be applied to the plant given a state value, $\mathbf{x}$, and a constant reference input, $\mathbf{r}$. Then, in principle, an artificial neural network (ANN) can be trained to reproduce an estimate, $\tilde{\mathbf{u}}^o$, of $\mathbf{u}^o$ if it is presented with $\mathbf{x}$ and $\mathbf{r}$. Now this process could be repeated with the *same* ANN so that it is capable of reproducing several *different* optimal control values previously calculated off line for different plant states and reference inputs when presented with the same plant states and reference inputs.

Since digital processors operate in discrete time, the optimal control computational methods actually yield a set of states,

$$\left\langle \mathbf{x}^k \right\rangle = \left[\mathbf{x}^k(t_o),\, \mathbf{x}^k(t_1),\ldots,\mathbf{x}^k(t_{N-1})\right] \qquad (51)$$

and a corresponding set of optimal controls,

$$\left\langle \mathbf{u}^{ok} \right\rangle = \left[\mathbf{u}^{ok}(t_o),\, \mathbf{u}^{ok}(t_1),\ldots,\mathbf{u}^{ok}(t_{N-1})\right] (52)$$

at N discrete times, $t_0,t_1,\ldots,t_{N-1}$, for a constant reference input, r^k, where the superscript, *k*, refers to the particular reference input. Let the complete set of results of these off line computations over a range of R different reference inputs be

$$\langle \mathbf{x} \rangle = \left\langle \left\langle \mathbf{x}^1 \right\rangle, \left\langle \mathbf{x}^2 \right\rangle, \ldots \left\langle \mathbf{x}^R \right\rangle \right\rangle \qquad (53)$$

and

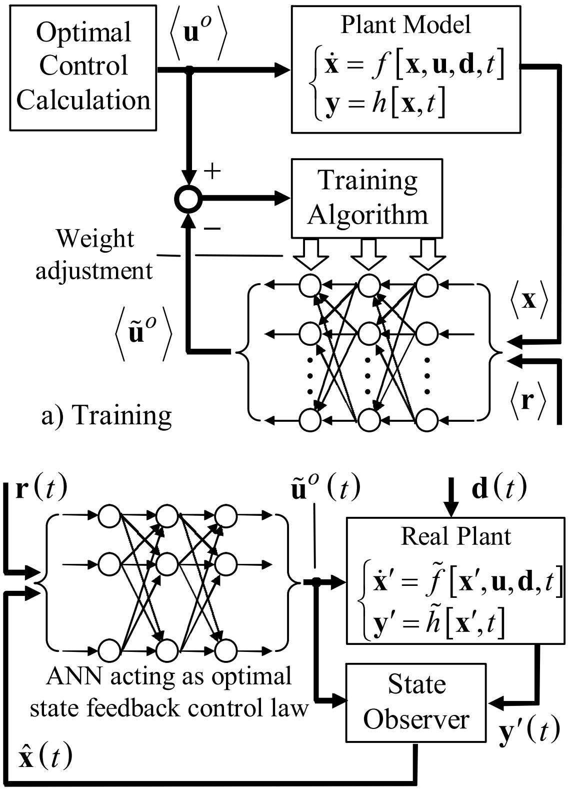
$$\left\langle \mathbf{u}^o \right\rangle = \left\langle \left\langle \mathbf{u}^{o1} \right\rangle, \left\langle \mathbf{u}^{o2} \right\rangle, \ldots \left\langle \mathbf{u}^{oR} \right\rangle \right\rangle \qquad (54)$$

Then, in principle, a suitable ANN could reproduce an estimate, $\left\langle \tilde{\mathbf{u}}^o \right\rangle$, of $\left\langle \mathbf{u}^o \right\rangle$, given $\langle \mathbf{x} \rangle$ and $\langle \mathbf{r} \rangle = \left\langle \mathbf{r}^0, \mathbf{r}^1, \ldots, \mathbf{r}^R \right\rangle$. The overall result would be that the ANN could be used to directly close the loop with an approximation to control law (50):

$$\tilde{\mathbf{u}}^o = \tilde{\mathbf{G}}^o(\mathbf{x},\mathbf{r}) \qquad (55)$$

Fig.. 2 summarises the training and implementation stages.

The scheme depends upon the important property of interpolation possessed by ANNs. The process of training an ANN to a number of discrete points, as explained above, results in a form of continuous multidimensional curve fitting. If the ANN is presented with a state value, $\mathbf{x}$, and a reference input value, $\mathbf{r}$, *not included in the original training set* but *lying inside the multidimensional polygonal surface encompassing the training set,* then it should produce a control value, $\tilde{\mathbf{u}}^o$, that is a close approximation to the correct optimal control value, $\mathbf{u}^o$. The success of this scheme would, of course, depend on the type of ANN selected and the number of neurons in the hidden layers. The reader may refer to texts such as (Sunan, et. al., 2004) or (Picton, 2000), to gain insight into the various types of ANN.


Optimal Control Calculation
$\langle \mathbf{u}^o \rangle$
Plant Model
$\begin{cases}\dot{\mathbf{x}} = f[\mathbf{x},\mathbf{u},\mathbf{d},t] \\ \mathbf{y} = h[\mathbf{x},t]\end{cases}$
+
Training Algorithm
−
Weight adjustment
$\langle \mathbf{x} \rangle$
$\langle \tilde{\mathbf{u}}^o \rangle$
$\langle \mathbf{r} \rangle$
$\mathbf{r}(t)$
$\tilde{\mathbf{u}}^o(t)$
$\mathbf{d}(t)$
Real Plant
$\begin{cases}\dot{\mathbf{x}}' = \tilde{f}[\mathbf{x}',\mathbf{u},\mathbf{d},t] \\ \mathbf{y}' = \tilde{h}[\mathbf{x}',t]\end{cases}$
ANN acting as optimal state feedback control law
State Observer
$\mathbf{y}'(t)$
$\hat{\mathbf{x}}(t)$

a) Training

b) Closed loop implementation

Fig. 2. Application of ANN for optimal state control.

6. Conclusions and Recommendations:

The comparison of the two classical open loop approaches to solving the optimal control problem is interesting. The dynamic-programming method yielded one nonlinear partial differential equation (15) to be solved while applying the maximum-principle to the same problem yielded two nonlinear ordinary differential equations (32) and (35). Although both techniques generate equations which require the aid of digital computers for solution, the two first-order ordinary differential equations from the maximum-principle are considered easier to solve than the nonlinear partial differential equation. In general, therefore Pontryagin's method is recommended, but Bellman's method should still be considered as in particular cases it might prove to be more straightforward.

In order to determine the attainable closeness to optimality, it is recommended that the new scheme depicted in Fig. 2 is first tested with very simple well known cases for which exact closed loop optimal controllers already exist, such as the double integrator plant. The next recommended step is to test out the scheme depicted in Fig 2 for the simple minimum energy control problem presented in section 4. The variation of the closeness to optimality with the number of training points and their distribution within the operating envelope of the plant should be investigated.

7. References:

Bellman R., *Dynamic Programming,* Princeton University Press, Princeton, NJ, 1957.

Bellman, R., Dreyfus S. E., *Applied Dynamic Programming*, Princeton University Press, Princeton, NJ, 1962.

Bolttyanskii, V. G., Gamkrelidze, R.V., Pontryagin, L. S, *The Mathematical Theory of Optimal Processes*, *I. The Maximum Principle*, Izv., Akad., Nauk, SSR, Ser. Mat. 24, 3, 1960.

Pearson A. B., "Synthesis of a Minimum Energy Controller subject to Average Power Constraint", *Joint Automatic Control Conference Proceedings*, New York, 1962, pp. 19-4-1 to 19-4-6.

Picton P., *Neural Networks*, Palgrave, 2000.
Pontryagin L. S., *Optimal Control Processes*, Usp. Mat. Nauk 14, 3, 1959.

Optimal Relay and Saturating Control System Synthesis Ryan, E. P., *Optimal Relay and Saturating Control System Synthesis*, P. Peregrinus for the IEE, 1982.

Shinners, S. M., *Modern Control System Theory and Design*, John Wiley & Sons, pp 632-668, 1992.

Sunan, H., Kok K. and Kok Z, *Neural Network Control: Theory and Application,* Research Studies Press Ltd., 2004.