Proceedings of the World Congress on Engineering 2007 Vol II
WCE 2007, July 2 - 4, 2007, London, U.K.

# Neural Networks for Optimal Control of Aircraft Landing Systems

Kevin Lau, Roberto Lopez and Eugenio Oñate *

*Abstract*—**In this work we present a variational formulation for a multilayer perceptron neural network. With this formulation any learning task for the neural network is defined in terms of finding a function that is an extremal for some functional. Thus the multilayer perceptron provides a direct method for solving general variational problems.**

**The application of this numerical method is investigated through an optimal control example, the aircraft landing problem. Using a multilayer perceptron neural network, the optimal control of the aircraft was determined by locating the extremal value of a variational problem formulated using the state variables of the aircraft.**

***Keywords: neural networks, multilayer perceptron, optimal control, aircraft landing.***

## 1 Introduction

Many problems arising in science and engineering aim to determine a function that is the optimal value of a specified functional. A functional is defined as a correspondence that assigns a number to each function belonging to some class. The extremal values of a functional can be determined using a branch of mathematics known as the calculus of variations [8]. Problems of this type are labelled variational problems.

Optimal control problems fall into the general class of variational problems, which are becoming increasingly more important in the design of modern engineering systems. In this type of problem the objective is to determine the input to a system which optimises a given objective functional, whilst satisfying a set of constraints on the input and the states of the system [10]. The control input that yields an extremum of the objective functional is known as the optimal control and the corresponding variation of the state variables is called the optimal trajectory.

While some simple optimal control problems can be solved analytically, general optimal control problems can only be solved by approximating the solution using direct methods [3]. The fundamental principle of this method is to reduce a variational problem into a function optimisation problem in many dimensions.

Here we present a variational formulation for the multilayer perceptron. In this formulation the learning task for the neural network consists of finding a function that is an extremal for some functional [12]. Thus the multilayer perceptron can be considered as a direct method for solving general variational problems, and consequently optimal control problems [11].

The application of this numerical method to the solution of optimal control problems is investigated by studying an aerospace example. Here we seek to determine the optimal control and the corresponding optimal trajectory of an aircraft during its final approach before landing. The aircraft landing problem examined here is similar to that considered in [6].

## 2 The aircraft landing problem

The landing of an aircraft consists of two main stages: the glide-path phase and the flare-out phase. In the first stage an air traffic controller guides the pilot to a position where the aircraft will be range of the Instrument Landing System (ILS). At this position the pilot sets the navigational receivers to the correct ILS frequency [5]. On board the aircraft a glide slope indicator utilises this ILS signal to give the pilot a visual indication if the aircraft is currently above or below the desired glide path.

At approximately 30 meters above the runway the pilot begins the second and final stage of landing, the flare-out procedure. At this point the ILS glide slope becomes unsuitable and the pilot must guide the aircraft along the desired trajectory by making visual contact with the ground [6]. It is assumed that at the beginning of this flare-out phase the values of the altitude and the altitude rate of the aircraft lie within a given range and that the aircraft has been guided to the correct position by the air traffic controller. It is also assumed that during the flare-out the aircraft is not affected by wind gusts or other perturbations.

In this work our attention is focussed on the final phase of the landing process, the flare-out. The aim of this

*International Center for Numerical Methods in Engineering (CIMNE), Edificio C1, Gran Capitan s/n, 08034 Barcelona, Spain. E-mail: kevinlau@cimne.upc.edu, rlopez@cimne.upc.edu, onate@cimne.upc.edu.

problem is to determine the optimal elevator deflection angle as a function of time, which also satisfies a set of performance requirements. The problem has been simplified as it is assumed that only the longitudinal motion of the aircraft need be considered, as the lateral motion is primarily used to set the orientation of the aircraft in the same direction as the runway, prior to the flare-out phase.

The longitudinal dynamics of the aircraft are controlled by the pitch angle via the elevator, a rotatable trailing edge flap traditionally located on the horizontal stabiliser [1]. The pitch angle of the aircraft is a measure of the degree the nose of the aircraft make with the earth in the vertical plane. By changing the elevator angle the pitching moment around the centre of mass of the aircraft is altered, causing a change in the pitch angle. Figure 1 depicts the elevator deflection angle and the pitch angle of an aircraft.

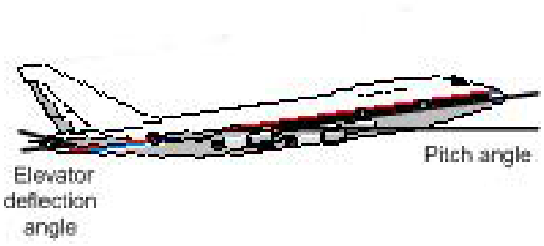


Figure 1: Elevator deflection angle and pitch angle.

## 2.1 State equations

Using the simplified conditions stated previously and the assumption that the glide angle is small ($\gamma \approx 0$), the equations of motion of the aircraft can be reduced to a group of equations known as the short period equations of motion [2].

These equations can be written using a set of transfer functions that relate the dynamic properties of aircraft and the control input to the state variables, as shown in Equation (1). The variables of the aircraft motion used in this equation are the pitch angle rate ($\theta'$), the altitude ($h$) and the altitude acceleration ($h''$). The control variable is the elevator deflection angle ($\delta$).

The properties of the aircraft are defined using a set of parameters defined as the short period gain ($K_s$), the short period resonant frequency ($\omega_s$), the short period damping factor ($\eta$), the path time constant ($T_s$) and conversion factor ($C_F$) [6]. The notation $'$ is used to denote the time derivative.

$$\begin{aligned}
\theta'(s) &= \frac{K_s(T_s s+1)}{\left(\frac{s^2}{\omega_s^2}+\frac{2\eta s}{\omega_s}+1\right)}\,\delta(s),\\
h''(s) &= \frac{C_F V}{T_s s+1}\,\theta'(s),\\
h(s) &= \frac{1}{s^2}\,h''(s),\\
h(s) &= \frac{C_F K_s V}{s^2\left(\frac{s^2}{\omega_s^2}+\frac{2\eta s}{\omega_s}+1\right)}\,\delta(s). \qquad (1)
\end{aligned}$$

The variable $V$ is the velocity of the aircraft, it is assumed to be constant during the flare-out phase at a value of $78\,ms^{-1}$. The aircraft parameters are also assumed to be time invariant, the numerical values used here are $K_s = -0.95\,s^{-1}$, $T_s = 2.5\,s$, $\omega_s = -0.95\,rad\,s^{-1}$, $\eta = 0.5$ and $C_F = 0.3048$.

However, some of the state variables used in Equation (1) are not readily available. For example the term $h''$ can be difficult to obtain. For this reason an alternate set of state variables will be used instead to describe the dynamics of the aircraft [6]:

$\theta'$, the pitch angle rate.
$\theta$, the pitch angle.
$h'$, the altitude rate.
$h$, the altitude.

This set of variables can be easily obtained from gyroscopic or radar altimeter measurements in flight. Equation (1) is then transformed into the following set of ordinary differential equations in the time domain,

$$\begin{aligned}
\frac{d\theta'}{dt} &= \left(\frac{1}{T_s}-2\eta\omega_s\right)\theta'(t)\\
&+ \left(\frac{2\eta\omega_s}{T_s}-\omega_s^2-\frac{1}{T_s^2}\right)\theta(t)\\
&+ \frac{1}{C_F}\left(\frac{1}{VT_s^2}-\frac{2\eta\omega_s}{VT_s}+\frac{\omega_s^2}{V}\right)h'(t)\\
&+ \omega_s^2 K_s T_s \delta(t),\\
\frac{d\theta}{dt} &= \theta'(t),\\
\frac{dh'}{dt} &= C_F\left(\frac{V}{T_s}\theta(t)-\frac{1}{T_s}h'(t)\right),\\
\frac{dh}{dt} &= h'(t). \qquad (2)
\end{aligned}$$

It can be seen that the elevator deflection angle ($\delta$) has a direct effect on the pitch angle rate ($\theta'$), which in turn affects the pitch angle ($\theta$), the altitude rate ($h'$) and the altitude ($h$).

### 2.2 Performance requirements

The performance requirements define the physical constraints and desired values of the control and the state variables. The most important of these are highlighted in the following section. In this example the flare-out procedure begins at $t_i = 0s$, and ends at the final or touchdown time $t_f = 20s$. The initial conditions of the different state variables are displayed in Table 1.

| | |
|---|---|
| $h_0$ | $30\,m$ |
| $h'_0$ | $6\,m\,s^{-1}$ |
| $\theta_0$ | $-0.078\,rad$ |
| $\theta'_0$ | $0\,rad\,s^{-1}$ |

Table 1: Initial Conditions.

The performance of the control input is evaluated by comparing the trajectories of the state variables against their desired variation. The desired altitude is given by the following expression

$$h_d(t) = \begin{cases} 30\exp\left(-\frac{t}{5}\right), & 0 \leq t \leq 15, \\ 6 - 0.3t, & 15 \leq t \leq 20. \end{cases} \quad (3)$$

This exponential-linear trajectory ensures a safe and comfortable landing. The desired altitude is displayed in Figure 2.

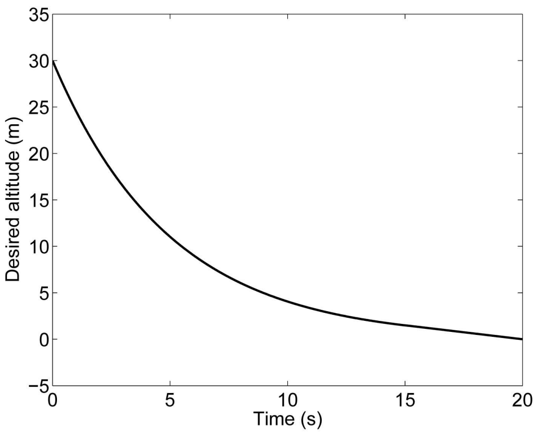


Figure 2: Desired altitude.

An important requirement is that the aircraft does not land before the final touchdown time, $t_f$. At this time the following condition must be satisfied

$$h_d(t_f) = 0. \quad (4)$$

The desired altitude rate, or vertical velocity, of the aircraft is the time derivative of Equation (3), and is given as

$$h'_d(t) = \begin{cases} 6\exp\left(-\frac{t}{5}\right), & 0 \leq t \leq 15, \\ -0.3, & 15 \leq t \leq 20. \end{cases} \quad (5)$$

At the time of touchdown the pitch angle of the aircraft must lie in the range $[0°, 10°]$. This requirement is defined by physical limitations. The lower limit serves to ensure the nose wheel of a tricycle landing gear does not touchdown prematurely. Similarly, the upper limit is set to prevent the tail gear touching downing first. A desired pitch angle at touchdown could be specified as

$$\theta_d(t_f) = 2°. \quad (6)$$

In order to ensure safety and comfortability during the landing phase, it is desirable to restrict the pitch angle rate from excessive fluctuations. Thus the desired pitch angle rate can be written as

$$\theta'_d(t) = 0, \quad 0 \leq t \leq t_f. \quad (7)$$

As stated earlier, the elevator controls the longitudinal motion of the aircraft. It is assumed here that any control signal is instantaneously represented by the elevator. The elevator deflection angle is also physically limited to a finite range

$$-35° \leq \delta(t) \leq +15°, \quad 0 \leq t \leq t_f. \quad (8)$$

Finally, it is desirable to land without expending excessive amounts of control effort. Therefore the desired elevator deflection angle can be defined as

$$\delta_d(t) = 0, \quad 0 \leq t \leq t_f. \quad (9)$$

## 3 A variational formulation for the multilayer perceptron

In this section we formulate the learning problem in the multilayer perceptron from a variational point of view [12]. This formulation provides a direct method to approximate the solution of any variational problem, and consequently any optimal control problem [11]. The steps required to solve a variational problem are shown in Figure 3, and are described in the following sections.

### 3.1 The multilayer perceptron function space

A neuron model is the basic information processing unit within a neural network; in the multilayer perceptron the characteristic neuron model used is the perceptron [14].

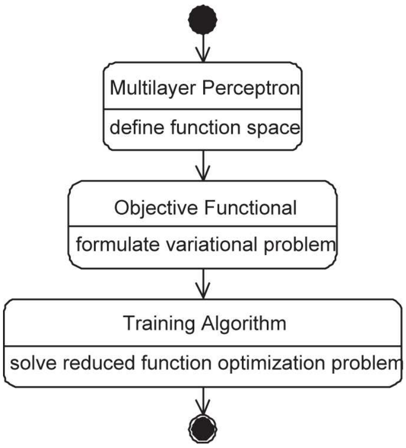


Figure 3: Activity Diagram.

A neural network is formed by connecting neurons together in a structure known as a network architecture. The network architecture used in the multilayer perceptron is feed-forward [14]. Thus a multilayer perceptron can be defined as a feed-forward network architecture composed of perceptron neuron models.

Mathematically, a multilayer perceptron spans a parameterised function space $V$ from an input $X \subseteq \mathbf{R}^n$ to an output $Y \subseteq \mathbf{R}^m$ [12]. Elements of $V$ are parameterised by the free parameters in the network, which can be grouped together in a $s$-dimensional free parameter vector $\underline{\alpha} = (\alpha_1, ..., \alpha_s)$. The dimension of the function space $V$ is therefore $s$. The elements of the function space spanned by a multilayer perceptron are of the form

$$\begin{aligned} \mathbf{y} : \mathbf{R}^n &\to \mathbf{R}^m \\ \mathbf{x} &\mapsto \mathbf{y}(\mathbf{x}; \underline{\alpha}). \end{aligned}$$

A multilayer perceptron with as few as one hidden layer of sigmoid neurons and an output layer of linear neurons provides a general framework for approximating any function from one finite dimensional space to another, provided a sufficient number of hidden neurons are available. Therefore, multilayer perceptron networks are a class of universal approximators [9].

### 3.2 The variational problem

Traditionally the learning problem for the multilayer perceptron has been formulated in terms of the minimisation of an error function of the free parameters, fitting the neural network to some input-target data set [4]. Thus the only possible learning tasks for the multilayer perceptron are data modeling type problems. In a variational formulation for the multilayer perceptron, the concept of error function, $e(\underline{\alpha})$, is changed to the concept of objective functional, $F[\mathbf{y}(\mathbf{x}; \underline{\alpha})]$ [12]. An objective functional for the multilayer perceptron is of the form

$$\begin{aligned} F : \quad V &\to \mathbf{R} \\ \mathbf{y}(\mathbf{x}; \underline{\alpha}) &\mapsto F[\mathbf{y}(\mathbf{x}; \underline{\alpha})]. \end{aligned}$$

The objective functional defines the task that the neural network is required to accomplish and provides a measure of the quality of the representation that it is required to learn. The choice of a suitable objective functional depends on the particular application. As shown in this work, changing the concept of an error function to the concept of an objective functional extends learning tasks for the multilayer perceptron to any variational problem.

The learning problem for the multilayer perceptron can be formulated in terms of the minimisation of an objective functional of the function space spanned by the neural network [12]:

*Let $V$ be the space of all functions $\mathbf{y}(\mathbf{x}; \underline{\alpha})$ spanned by a multilayer perceptron, and let $s$ be the dimension of $V$. Find a function $\mathbf{y}^*(\mathbf{x}; \underline{\alpha}^*) \in V$ for which the functional $F[\mathbf{y}(\mathbf{x}; \underline{\alpha})]$, defined on $V$, takes on a minimum or a maximum value.*

A variational problem for the multilayer perceptron can be specified by a set of constraints, which are equalities or inequalities that the solution must satisfy. Such constraints are expressed as functionals. A simple approach is to unconstrain the constrained problem by adding a penalty term to the original objective function for each constraint in the problem.

### 3.3 The reduced function optimisation problem

The objective functional, $F[\mathbf{y}(\mathbf{x}; \underline{\alpha})]$, has an objective function associated, $f(\underline{\alpha})$, which is defined as a function of the free parameters in the network [12],

$$\begin{aligned} f : \mathbf{R}^s &\to \mathbf{R} \\ \underline{\alpha} &\mapsto f(\underline{\alpha}). \end{aligned}$$

The minimum or maximum value of the objective functional is achieved with a vector of free parameters at which the objective function takes on a minimum or maximum value, respectively. Therefore the learning problem for the multilayer perceptron, formulated as a variational problem, can be reduced to a function optimisation problem [12]:

ISBN:978-988-98671-2-6 

*Let $\mathbf{R}^s$ be the space of all vectors $\underline{\alpha}$ spanned by the free parameters of a multilayer perceptron. Find a vector $\underline{\alpha}^* \in \mathbf{R}^s$ for which the function $f(\underline{\alpha})$, defined on $\mathbf{R}^s$, takes on a minimum or a maximum value.*

In this sense, a variational formulation for the multilayer perceptron provides a direct method to approximate the solution of general variational problems, in any dimension and up to any desired degree of accuracy [12].

The training algorithm is entrusted to solve the reduced function optimisation problem. There are several different training algorithms available for the multilayer perceptron, some of the most widely used are the conjugate gradient [15], the quasi-Newton method [15] and the evolutionary algorithm [7].

## 4 Problem solution

In this section a multilayer perceptron is trained to determine the optimal control input for the aircraft landing problem, as formulated in Section 2. The problem is solved using the Flood library [13].

The first step in solving this problem is to choose the network architecture, in order to define a function space for the control variable. Here a multilayer perceptron with a sigmoid hidden layer and a linear output layer is used. This neural network is a class of universal approximator [9]. Figure 4 is a graphical representation of the network architecture used to solve this problem.

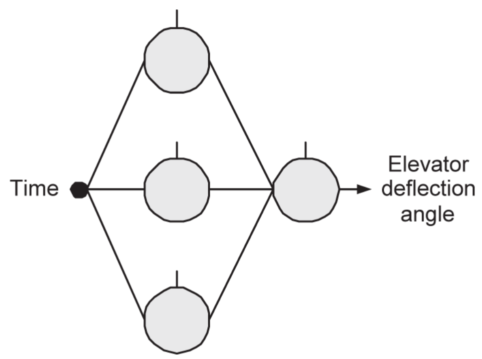


Figure 4: Network architecture for the aircraft landing problem, with one input, three neurons in the hidden layer and one output neuron.

The chosen neural network has one input, the time, and one output neuron representing the elevator deflection angle. An initial guess of the number of neurons in the hidden layer was taken as three. This neural network can be denoted as a $1:3:1$ multilayer perceptron. Such a multilayer perceptron spans a family $V$ of parameterised functions $\delta(t;\underline{\alpha})$ of dimension $s = 10$, which is the number of free parameters in the neural network. Elements $V$ are of the form

$$\begin{aligned} \delta: \quad \mathbf{R} &\rightarrow \mathbf{R} \\ t &\mapsto \delta(t;\underline{\alpha}), \end{aligned}$$

where

$$\delta(t;\underline{\alpha}) = b_1^{(2)} + \sum_{j=1}^{3} w_{1j}^{(2)} \cdot \tanh\left(b_j^{(1)} + w_{j1}^{(1)} t\right). \qquad (10)$$

The free parameters of the neural network are initialised such that the elevator deflection angle $\delta(t;\underline{\alpha})$ is 0 for all landing time $0 \leq t \leq t_f$. The elevator deflection angle must also be constrained to lie in the range $[-35°, +15°]$; the output of the neural network is therefore bounded as follows

$$\delta(t;\underline{\alpha}) = \begin{cases} -35°, & \delta(t;\underline{\alpha}) < -35°. \\ \delta(t;\underline{\alpha}), & -35° \leq \delta(t;\underline{\alpha}) \leq +15°. \\ +15°, & \delta(t;\underline{\alpha}) > +15°. \end{cases} \qquad (11)$$

The second step is to select a suitable objective functional in order to formulate the variational problem. This functional will determine the form of the optimal control function ($\delta^*(t)$), and is based upon the performance requirements discussed in Section 2.2.

From Equations (3) (4), (5), (6), (7) and (9) the objective functional used in this problem is defined as

$$\begin{aligned} F[\delta(t;\underline{\alpha})] &= \alpha_h \int_0^{t_f} [h(t) - h_d(t)]^2 dt \\ &+ \beta_h h_d(t_f) \\ &+ \alpha_{h'} \int_0^{t_f} [h'(t) - h'_d(t)]^2 dt \\ &+ \beta_\theta \theta_d(t_f) \\ &+ \alpha_{\theta'} \int_0^{t_f} [\theta'(t) - \theta'_d(t)]^2 dt \\ &+ \alpha_\delta \int_0^{t_f} [\delta(t) - \delta_d(t)]^2 dt. \end{aligned} \qquad (12)$$

where $\alpha_h$, $\alpha_{h'}$, $\alpha_{\theta'}$ and $\alpha_\delta$ are the altitude, altitude rate, pitch angle rate and elevator deflection angle weight factors; the terms $\beta_h$ and $\beta_\theta$ are the touchdown weight factors for the altitude and the pitch angle. Table 2 displays the values used in this investigation. These numbers are design variables of the problem and they were obtained with some trial and error.

Note that evaluation of the objective functional, Equation (12), requires the time history of all the state variables in response to the time history of the control variable. These are determined by numerically integrating

| | |
|---|---|
| $\alpha_h$ | $5.0 \times 10^1$ |
| $\beta_h$ | $5.0 \times 10^3$ |
| $\alpha_{h'}$ | $1.0 \times 10^{-4}$ |
| $\beta_\theta$ | $2.0 \times 10^6$ |
| $\alpha_{\theta'}$ | $2.0 \times 10^6$ |
| $\alpha_\delta$ | $1.0 \times 10^5$ |

Table 2: Weight factors.

the state equations of the system using the Runge-Kutta method with 1000 integration points [15]. The objective functional also requires numerical integration for evaluation; in this problem the trapezoid method has been used.

The third step is to choose a suitable training algorithm to solve the reduced function optimisation problem. Here a quasi-Newton method with BGFS train direction and Brent optimal train rate methods have been used [15]. The tolerance in the Brent's method is set to $10^{-6}$. The objective function gradient vector $\nabla f(\underline{\alpha})$ is also evaluated using numerical differentiation. Here the symmetrical central differences method has been used, with an epsilon value of $10^{-6}$ [4].

The training algorithm is set to stop when the optimal train rate in Brent's method reaches 0. In our example case, the quasi-Newton method required 291 epochs or iterations to find the minimum of the objective functional. The evaluation of the initial guess was 4.76; after training this value fell to $1.002 \times 10^{-4}$. Figure 5 displays the training history of this problem, with the objective functional evaluation plotted against the number of training epochs. Note that the Y-axis uses a logarithmic (base 10) scale.

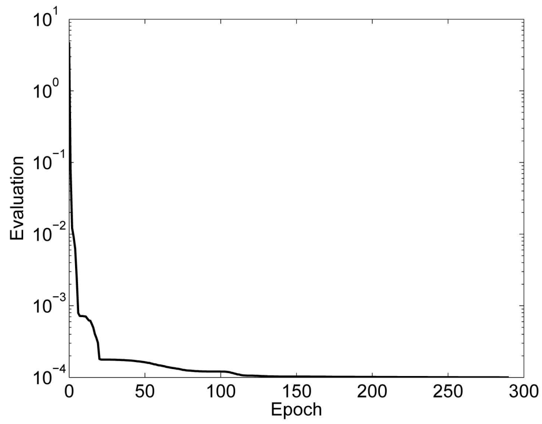


Figure 5: Training history.

The training results of this problem are displayed in Table 3. Here $N$ denotes the number of epochs, $M$ the number of objective function evaluations, $f(\underline{\alpha}^*)$ the final objective function value and $\|\nabla f(\underline{\alpha}^*)\|$ the final objective function gradient norm. From this table it can be seen that the final gradient norm approaches a very small value, which indicates that the training algorithm has converged to a minimum point.

| | |
|---|---|
| $N$ | 291 |
| $M$ | 16748 |
| $f(\underline{\alpha}^*)$ | $1.002 \times 10^{-4}$ |
| $\|\nabla f(\underline{\alpha}^*)\|$ | $6.09 \times 10^{-2}$ |

Table 3: Training results.

The optimal control obtained by the neural network is shown in Figure 6.

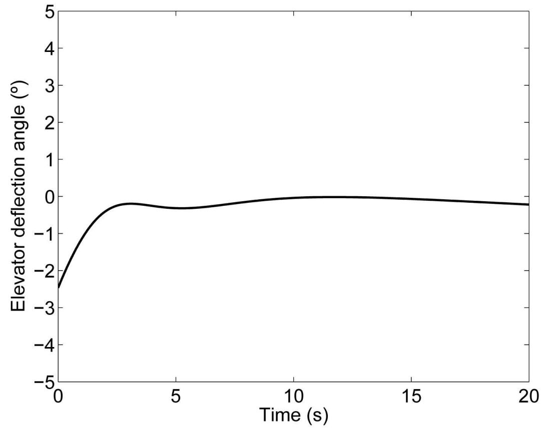


Figure 6: Optimal control (elevator deflection angle) for the aircraft landing problem.

Figure 6 shows the elevator deflection angle during the landing phase. It can be seen that the magnitude of control input ranges from -3° to 0° and has a smooth profile. It can be seen that the effort required to control the aircraft has been minimised.

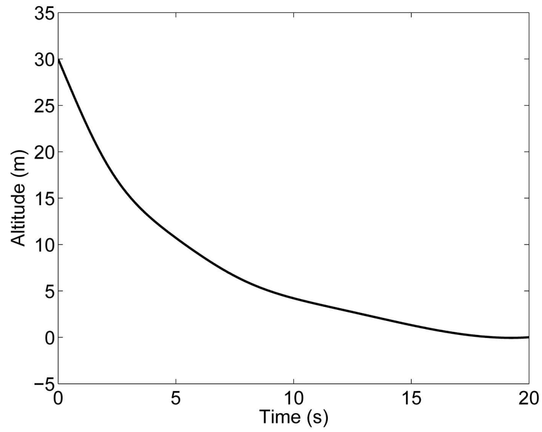


Figure 7: Optimal altitude trajectory.

ISBN:978-988-98671-2-6

The optimal trajectory for the altitude, depicted in Figure 7, matches the desired altitude profile detailed in Equation (3). It can be seen that at the final touchdown time, $t_f$, the altitude of the aircraft is $0\,m$.

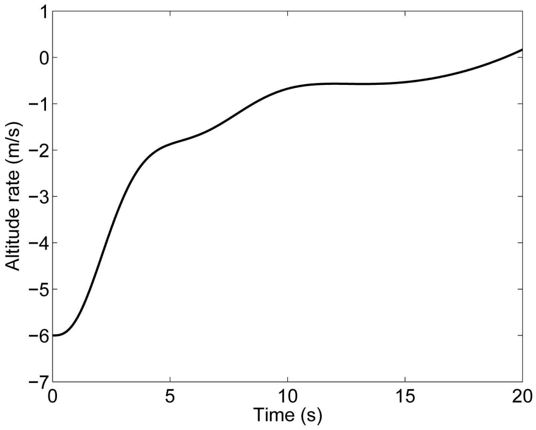


Figure 8: Optimal altitude rate trajectory.

Figure 8 shows that the altitude rate of the aircraft; it can be seen that it follows the profile specified by Equation (5).

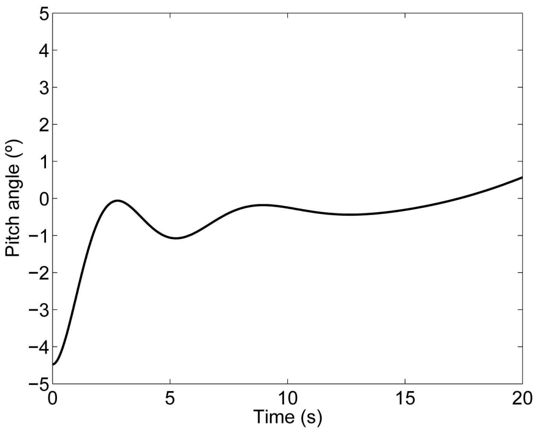


Figure 9: Optimal pitch angle trajectory.

The pitch angle variation during the landing phase is displayed in Figure 9. The value at touchdown is approximately $0.5°/s$. This value lies within the desired range $[0°, 10°]$. However the desired touchdown value of $2°$ has not been obtained. This is believed to be a result of the simplified state equations used; the model does not account for effects such as induced lift at low altitudes.

Finally, Figure 10 displays the pitch angle rate. It can be seen that throughout the landing phase the magnitude of the pitch angle rate is relatively small, and its profile is sufficiently smooth to ensure a comfortable and safe landing.

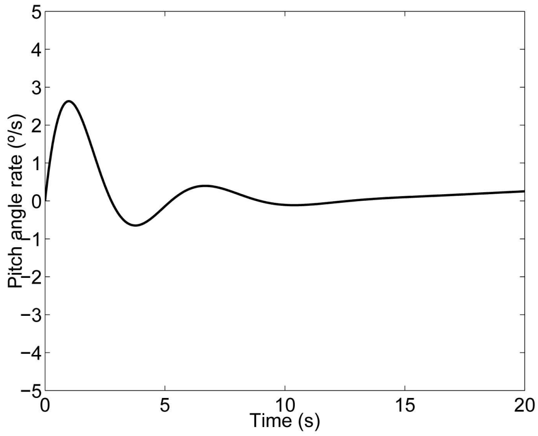


Figure 10: Optimal pitch angle rate trajectory.

## 5 Conclusions and future work

In this work the variational formulation of the multilayer perceptron has been used to provide a direct method for the solution of general variational problems. This numerical method has been applied to finding the optimal input control of an aircraft landing system. The corresponding state variables of the system have been shown to satisfy the performance requirements.

However, this application requires some further refinement. In particular, a more complete set of state equations describing the aircraft dynamics should be used.

## References

[1] Holt, A., *Engineering Analysis of Flight Vechicles*, Dover Publishing, 1992.

[2] Babister, A.W., *Aircraft Dynamic Stability and Response*, Pergamon Press, 1980.

[3] Betts, J.T., A survey of numerical methods for trajectory optimization, *AIAA Journal of Guidance, Control and Dynamics*, V21, N2, pp. 193-207, 1998.

[4] Bishop, C., *Neural Networks for Pattern Recognition*, Oxford University Press, 1995.

[5] *U.S. Centennial of Flight Commission*, www.centennialofflight.gov, 2006.

[6] Ellert, F.J., Merriam, C.W., *Synthesis of feedback controls using optimization theory - An example*, IEEE Transactions on Automatic Control, V8, N2, pp. 89-103, 1963.

[7] Fogel, D.B., An introduction to simulated evolutionary optimization, *IEEE Transactions on Neural Networks* V5, N1, pp. 3-14, 1994.

ISBN:978-988-98671-2-6

[8] Gelfand, I.M., Fomin, S.V., *Calculus of Variations*, Prentice Hall, 1963.

[9] Hornik, K., Stinchcombe, M., White, H., *Multilayer feedforward networks are universal approximators*, Neural Networks, V2, N5, pp. 359-366, 1989.

[10] Kirk, D.E., *Optimal Control Theory. An Introduction*, Prentice Hall, 1970.

[11] Lopez, R., Balsa-Canto, E., Oñate, E., Artificial neural networks or the solution of optimal control problems. *Proceedings of the 6th International Conference on Evolutionary and Deterministic Methods for Design, Optimisation and Control with Applications to Industrial and Societal Problems EUROGEN 2005*, 2005.

[12] Lopez, R., Oñate, E., A variational formulation for the multilayer perceptron. *Proceedings of the 16th International Conference on Artificial Neural Networks ICANN 2006*, 2006.

[13] Lopez, R., *Flood: An Open Source Neural Networks C++ Library*, www.cimne.com/flood, 2006.

[14] Šíma, J., Orponen, P., *General-Purpose Computation with Neural Networks: A Survey of Complexity Theoretic Results*, Neural Computation, V15, pp. 2727-2778, 2003.

[15] Press, W.H., Teukolsky, S.A , Vetterling, W.T., Flannery, B.P., *Numerical Recipes in C++: The Art of Scientific Computing*, Cambridge University Press, 2002.