MULTI-LAYER FUNCTIONAL APPROXIMATION OF NON-LINEAR UNSTEADY AERODYNAMIC RESPONSE

by

Flávio Donizeti Marques

Thesis submitted to the Faculty of Engineering,

University of Glasgow, for the Degree of Doctor of Philosophy

March 1997

Abstract

Non-linear unsteady aerodynamic effects present major modelling difficulties in the analysis of aeroelastic response and in the subsequent design of appropriate controllers. As the direct use of the basic fluid mechanic equations is still not practical for aeroelastic applications, approximate models of the non-linear unsteady aerodynamic response are required. A rigorous mathematical framework, that can account for the complex non-linearities and time-history effects of the unsteady aerodynamic response, is provided by the use of functional representations. A recent development, based on functional approximation theory, has provided a new functional form; namely, multi-layer functionals. Moreover, the multi-layer functional representation for time-invariant, infinite memory systems is shown to be realisable in terms of temporal neural networks.

In this work, a multi-layer functional representation of non-linear motion-induced unsteady aerodynamic response is presented. A discrete-time, finite memory temporal neural network, in the form of a finite impulse response (FIR) neural network, is used as a practical realisation of a multi-layer functional. This model form permits the identification of parametric input-output models of the non-linear motion-induced unsteady aerodynamic response. Identification of an appropriate FIR neural network model is facilitated by means of a supervised training process using multiple sets of motion-induced unsteady aerodynamic response. The training process is based on a conventional genetic algorithm to optimise the FIR neural network architecture, and is combined with a simplification of the simulated annealing

algorithm to update weight and bias values.

The identification process is used to produce FIR neural network models for two types of non-linear unsteady flow regimes. The first model relates to weakly non-linear behaviour of the unsteady aerodynamic response for mildly separated flow-fields as defined by a semi-empirical model. The second model relates to non-linear unsteady aerodynamic response in the transonic regime as defined by a CFD code based on solution of the Euler equations. Generally, the training process presents a satisfactory performance in both problems showing that the combination of genetic algorithms and temporal neural networks provides a suitable framework for the non-linear unsteady aerodynamic response modelling. The approach is shown to furnish a satisfactory generalisation property to different motion histories at different Mach numbers, considering that only limited training set data is presented during the identification process.

Acknowledgements

I am deeply indebted to Dr. John Anderson for his assistance, advice and enormous patience during the supervision work until the final preparation of this thesis.

I want to thank the Brazilian Federal Agency, *Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq, Brasília, DF, Brasil*, for the financial support during the tenure of my postgraduate research studies in the Department of Aerospace Engineering, University of Glasgow.

I would also like to express my gratitude to all members of staff and friends of the Department of Aerospace Engineering, University of Glasgow who directly and indirectly helped me complete my work; in particular, to Mr. Laurent Dubuc, Dr. Frank Coton and Dr. Ken Badcock for their assistance in the generation of aerodynamic data.

Contents

Nomenclature

$C_L(t)$ unsteady aerodynamic lift force coefficient response at time t;

$C_{m_{1/4}}(t)$ unsteady aerodynamic pitch moment coefficient response at 25% chord length at time t;

$C_N(t)$ unsteady aerodynamic normal force coefficient response at time t;

$F(t)$ generalised unsteady aerodynamic force response at time t;

$F_u[u(\xi);t,\tau]$ indicial generalised force response at time t per unit step change in u occuring at time τ (*cf.* Equation 1.7);

$\mathbf{F}(t)$ generalised unsteady aerodynamic force response vector at time t (*cf.* Equation 1.4);

H_i Hermite polynomials of i^{th}-degree;

I unit hypercube;

L total number of time samples per training set;

M Mach number;

N_c number of input-output training sets;

N_i^f chromosome flag indicating whether the neuron i exists or not (*cf.* Figure 3.3);

P_n probability for the mutation of a neuron value (existent or non-existent):

P_t probability for the mutation of a time-delay value;

T_{ji} memory span of the synapse i belonging to the neuron j:

U set of functions in a space of infinite dimension $C[p_1, p_2]$ (*cf.* Inequality 2.3);

U_∞ freestream velocity;

V_i Laguerre functions of i^{th}-order;

a and b scaling coefficients for the selection operator (*cf.* Table 3.1);

c number of times the best individual is expected to be selected for reproduction;

$d_k(n)$ desired output of training set k at discrete-time n;

f fitness function (*cf.* Equation 3.1);

h_i unit impulse response of process unit i;

h_i^V Volterra kernel of i^{th}-order (*cf.* Equation 1.9);

h_{ji} impulse response of neuron j due to excitation applied to synapse i;

i, j, k, ℓ, m, p, q integer valued auxiliary constants;

n discrete-time step;

$\mathbf{n}$ normal surface vector operator (*cf.* Equation 1.4);

$p(\phi(\mathbf{x}, t))$ pressure distribution (*cf.* Equation 1.4);

t time;

u_1 and u_2 generalised motion histories used in the formation of the indicial response (*cf.* Figure 1.4);

u_t scalar generalised coordinate or displacement history:

$\mathbf{u}(t)$ instantaneuous boundary generalised motion input vector at time t;

$\mathbf{u}_t$ generalised coordinate or displacement history vector;

v_j activation potential of neuron j;

$w_{ji}(\tau_{ji})$ weight value of synapse i belonging to neuron j corresponding to the time-delay τ_{ji};

$\mathbf{w}_{ji}$ weight vector of synapse i belonging to neuron j;

$x_i(t)$ excitation applied to synapse i at time t;

x_η function in the set of real valued continuous functions with domain I (*cf.* Equation 2.5);

$\mathbf{x}$ vector of the spatial coordinates (*cf.* Equation 1.4);

$y(t)$ dynamic system output response at time t;

Ω flow domains;

$\alpha(t)$ angle of attack value at time t;

α_t angle of attack history;

β perturbation constant used to update weight and bias values (*cf.* Equation 3.4);

$\delta\mathbf{u}$ virtual generalised displacement;

δ^*W virtual work (*cf.* Equation 1.3);

ε positive real valued constant;

ζ real valued constant;

η auxiliary variable running in the unit hypercube I (*cf.* Equation 2.5);

θ_i bias values of neuron i;

μ function in the set of functions of bounded variation on the unit hypercube I (*cf.* Equation 2.5);

ξ auxiliary time variable running form the initial time instant to time instant τ;

τ arbitrary time instant;

τ_{ji} number of time-delay units of the finite memory filter in synapse i belonging to the neuron j;

$\boldsymbol{\phi}(\mathbf{x}, t)$ vector of the spatio-temporal flow state variables (*cf.* Equation 1.1);

φ activation function of a neuron defined as a non-constant, bounded, monotone-increasing continuous function (for example, a sigmoidal function);

$\mathcal{A}$ linear affine functional representation (*cf.* Equation 2.4);

$\mathcal{B}$ boundary operator (*cf.* Equation 1.2);

$\mathcal{F}$ functional representation;

$\mathcal{L}$ linear functional representation;

$\mathcal{MF}$ multi-layer functional representation (*cf.* Equation 2.6);

$\mathcal{N}$ non-linear partial differential operator;

$\mathcal{U}$ boundary displacement function;

$\partial\Omega$ flow boundaries;

$\partial\Omega_S$ boundaries of the aerodynamic surface;

$[\cdot]^T$ transpose matrix operation;

$|\cdot|$ norm operator.

Acronyms

CFD **C**omputational **F**luid **D**ynamics;

FIR **F**inite **I**mpulse **R**esponse;

NACA **N**ational **A**dvisory **C**ommitte of **A**eronautics.

Chapter 1

Introduction

1.1 Modelling Unsteady Aerodynamic Behaviour in Aeroelastic Applications

Modelling unsteady aerodynamic behaviour presents a significant challenge for the prediction and control of adverse aeroelastic phenomena. The earliest literature on aeroelasticity; for example, Scalan & Rosenbaum [1951], Bisplinghoff & Ashley [1962], Dowell *et al.* [1989], Fung [1993], and Bisplinghoff *et al.* [1996], incorporated unsteady aerodynamic models based, primarily, on linear potential theory. Linearised models of this kind have proved satisfactory, mainly because in many practical problems, the unsteady flowfield is adequately described by small perturbations of a uniform, inviscid, and irrotational freestream. The fundamental flow conditions that can be described by linear theory, and the limitations of linearised aerodynamic models for the prediction and control of aeroelastic phenomena such as divergence, flutter, and gust response, are well understood from a fluid dynamics point of view.

With the enlargement of flight operational conditions in modern aviation, analysis of aeroelastic problems can no longer neglect non-linear effects for describing unsteady aerodynamic behaviour (Dowell & Ilgamov, 1988 and Dowell, 1993). In-

deed, complex non-linear effects are constantly present in modern aircraft flight regimes: for example, dynamic stall on helicopter blades, and excursion of shock waves over aircraft manoeuvring at transonic speeds.

Non-linear effects are difficult to predict or model, whatever the dynamic system in question. For unsteady aerodynamic modelling, the non-linear flow effects of interest are mostly due to separated flows and compressibility effects leading to the appearance and dynamic excursion of shock waves. Their modelling is particularly difficult because of the lack of complete understanding on some physical aspects of unsteady flows; for example, separation and turbulence mechanisms. Surveys on physical and modelling aspects of unsteady flow effects can be found in the works by McCroskey [1977, 1982], Tijdeman & Seebass [1980], Ericsson & Reding [1987] and Mabey [1989].

For aeroelastic applications, the ideal and, perhaps, most general aero-structural model would be based on solutions of the non-linear fluid mechanics equations, which considers unsteady, compressibility and viscous effects, simultaneously with the solution of the equations of motion. The instantaneous states, which are generated by each of the corresponding equations, would be exchanged and the global simultaneous solution would produce both aerodynamic response and structural motion histories, which depend on the given initial conditions. Figure 1.1 shows an illustration for this general approach to aeroelastic modelling.

The problem in applying the general aeroelastic model, as represented in Figure 1.1, is mainly related to the unsteady aerodynamic model in use. Solutions to the non-linear fluid mechanics equations have been the focus of a great amount of research effort. For practical applications, however, solutions of the general fluid mechanics equations can usually be attained only by means of numerical techniques, or computational fluid dynamics (CFD) methods (Edward & Thomas, 1989 and Anderson, 1991), that normally demand extensive computations. These methods

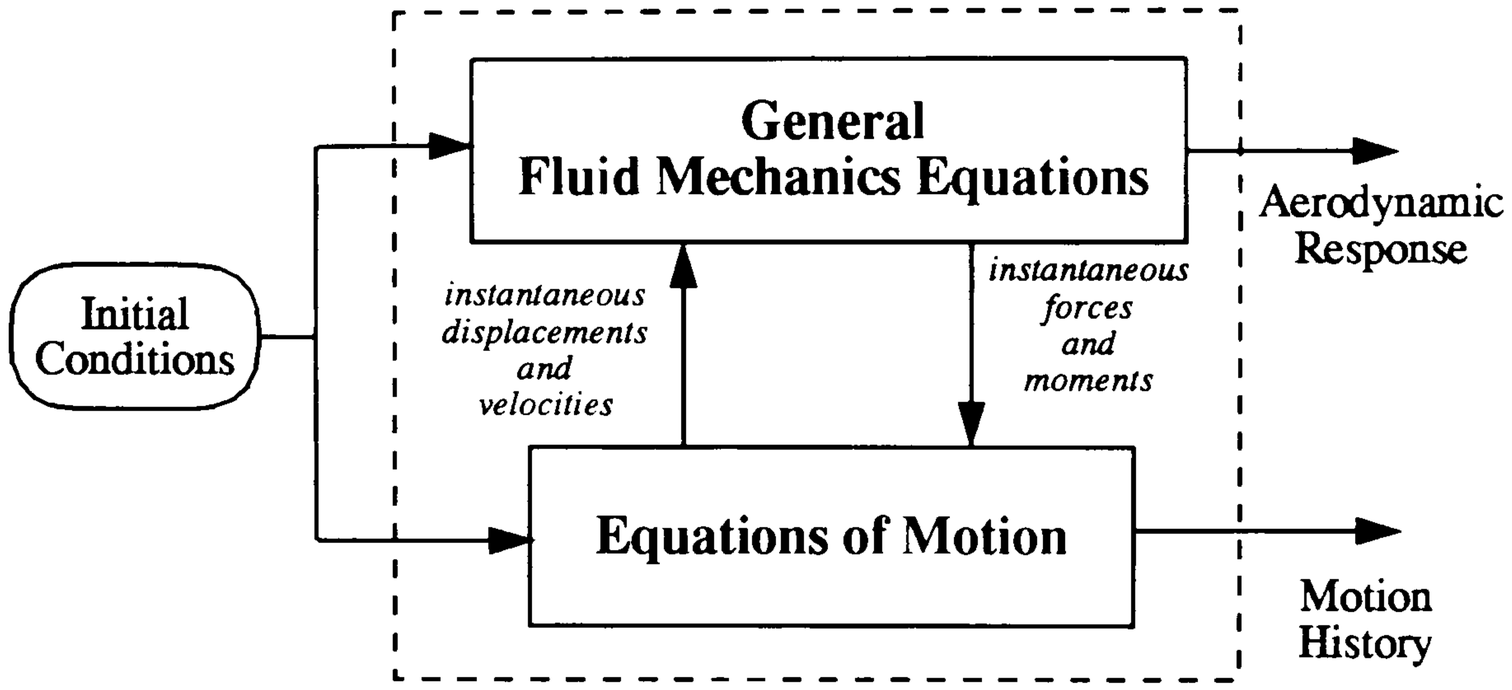


Figure 1.1: Representation of a general aeroelastic model.

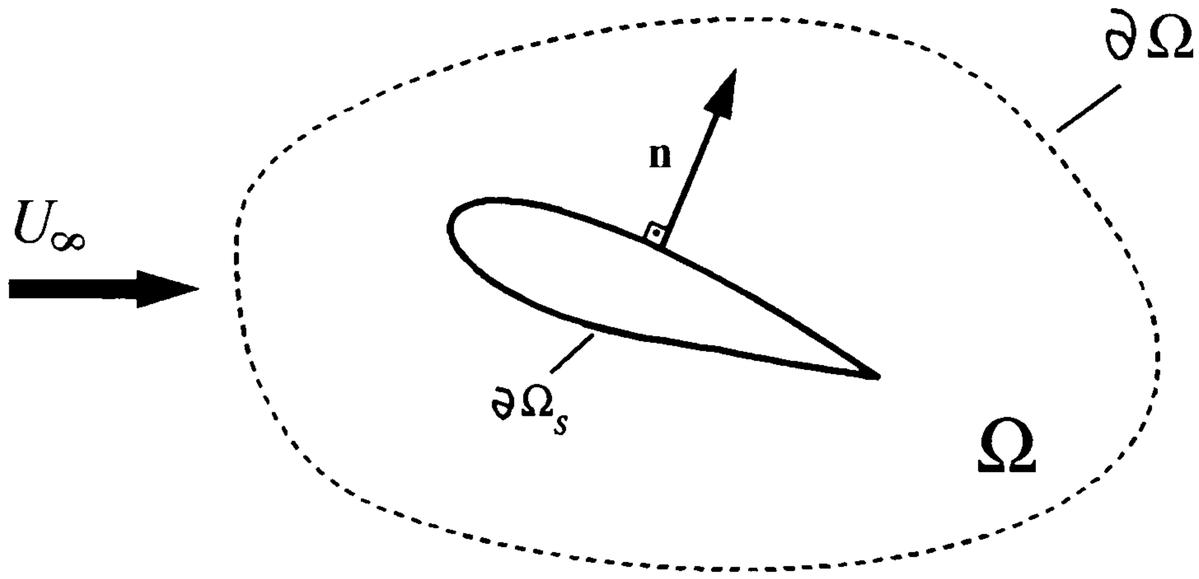
encompass any numerical technique for specific fluid mechanics applications. For instance, finite-difference, finite volume, and finite element techniques are frequently used in numerical solutions in fluid mechanics applications.

Limitations of CFD methods are basically the ones concerning the great amount of computations required. Consequently, CFD methods are still not appropriate for preliminary aeroelastic stability analysis and control design. Nevertheless, with the fast developments in computing and numerical techniques, CFD methods may be widely accessible in the near future, helping the convenient direct manipulation of fully non-linear fluid mechanics equations for aeroelastic analysis and control design. Alternatives for practical unsteady aerodynamic models to applications in aeroelasticity are, therefore, justified.

Alternative models of non-linear unsteady aerodynamics for aeroelastic applications have been achieved on the basis of some essential assumptions. Primarily, in aeroelastic models, the *decoupling* between the fluid mechanics equations and the equations of motion (*cf.* Figure 1.1) is an assumption that eliminates the need for simultaneous solution of the combined aero-structural set of equations. Therefore, by this premise the unsteady aerodynamic model is determined in isolation of the

physical laws governing the structural motion.

An intrinsic element of this decoupling process is that any alternative unsteady aerodynamic response model should account for the spatio-temporal behaviour of the internal aerodynamic states. For example, the decoupled unsteady inviscid fluid dynamic equations are described by the following symbolic representation.



$$\frac{\partial \boldsymbol{\phi}(\mathbf{x},t)}{\partial t} = \mathcal{N}\left(\boldsymbol{\phi}(\mathbf{x},t)\right) \quad \text{in } \Omega, \quad t > 0 \tag{1.1}$$

subject to

$$\mathcal{B}\left(\boldsymbol{\phi}(\mathbf{x},t), \mathbf{u}(t)\right) = 0 \quad \text{on } \partial\Omega_S \tag{1.2}$$

and an appropriate set of initial conditions, where, $\boldsymbol{\phi}(\mathbf{x},t)$ represents a vector of the spatio-temporal flow state variables, $\mathbf{x}$ is the vector of the spatial coordinates, $\mathcal{N}$ is a non-linear partial differential operator, Ω determines the flow domains of the problem, $\partial\Omega$ is the flow boundaries, $\partial\Omega_S$ represents the boundaries of the aerodynamic surface, $\mathcal{B}$ represents a boundary operator, and $\mathbf{u}(t)$ is the instantaneuous boundary generalised motion input vector.

By assuming boundary motions of the form $\mathcal{U}(\mathbf{x}, \mathbf{u}(t))$, such that the virtual work, $\delta^* W$, is defined by,

$$\delta^* W = \delta \mathbf{u}^T \mathbf{F}(t) \tag{1.3}$$

then the generalised unsteady aerodynamic force response vector, $\mathbf{F}(t)$ is given by the expression,

$$\mathbf{F}(t) = \int_{\partial\Omega_S} \left(\frac{\partial \mathcal{U}(\mathbf{x}, \mathbf{u}(t))}{\partial \mathbf{u}(t)} \right)^T \mathbf{n} \left(-p \left(\phi(\mathbf{x}, t) \right) \right) \, d\partial\Omega_S \tag{1.4}$$

where p is the pressure distribution described as a non-linear function of the instantaneous spatio-temporal flow variables and modified by the normal surface vector operator $\mathbf{n}$.

The basic assumption for unsteady aerodynamic models is that the influence of the implicit time-delays on pressure variations introduced by the spatio-temporal propagation and convection of flow variables can be represented by the motion history alone. Applying the principles of dynamic systems theory, unsteady aerodynamic models can be obtained from mathematical laws so that the generalised aerodynamic force response vector, $\mathbf{F}(t)$, can be represented as a non-linear functional, $\mathcal{F}$, of this generalised coordinate or displacement history, $\mathbf{u}_t$, illustrated in Figure 1.2 and described as,

$$\mathbf{F}(t) = \mathcal{F}[\mathbf{u}_t] \tag{1.5}$$

Figure 1.2: Schematic of dynamic systems.

Therefore, the functional representation in Equation 1.5 between unsteady aerodynamic response and motion history implicitly accounts for the effects of internal

flow states dynamics. Although this assumption leads to an exact functional representation of the linear unsteady aerodynamic response, for the non-linear case, this approach can only be used as an approximation.

Formal mathematical approaches to determine the functional relationship of the hereditary behaviour of unsteady aerodynamic responses, are originally due to the use of the superposition principle over transient responses to step changes, namely, *indicial responses* (Tobak & Pearson, 1964 and Etkin & Reid, 1996). This approach, which provides exact representation of the linear unsteady aerodynamic behaviour, is categorised as a *functional* due to its dependence on the complete (or partial) motion histories.

Viewing extensions to non-linear unsteady aerodynamic response functionals, Tobak and co-workers (Tobak & Pearson, 1964; Tobak & Schiff, 1978, 1981 and Tobak & Chapman, 1985) have proposed the indicial response functions to be reformulated as functionals of the motion histories. The result, after applying a generalisation of the superposition principle, is the non-linear unsteady aerodynamic response functional. However, practical use of the resulting complex integral equations is only permitted by simplifications as proposed by Tobak & Schiff [1978, 1981] and Jenkins [1991]; for example, by replacing the mathematical description of the motion history by its Taylor series expansion, or by assuming a limited dependence on the motion past values. Other functional forms; for instance, the Volterra series (Silva, 1993a, 1993b) also provide appropriate frameworks to the production of non-linear unsteady aerodynamic functionals.

Semi-empirical methods, or phenomenological models, comprise a class of aerodynamic models based on the premise of modelling unsteady flow response by considering its functional relationship with respect to the motion histories. Indeed, most of the knowledge on unsteady flow behaviour is due to experimental work, and basically, semi-empirical models use the information from these experiments to establish

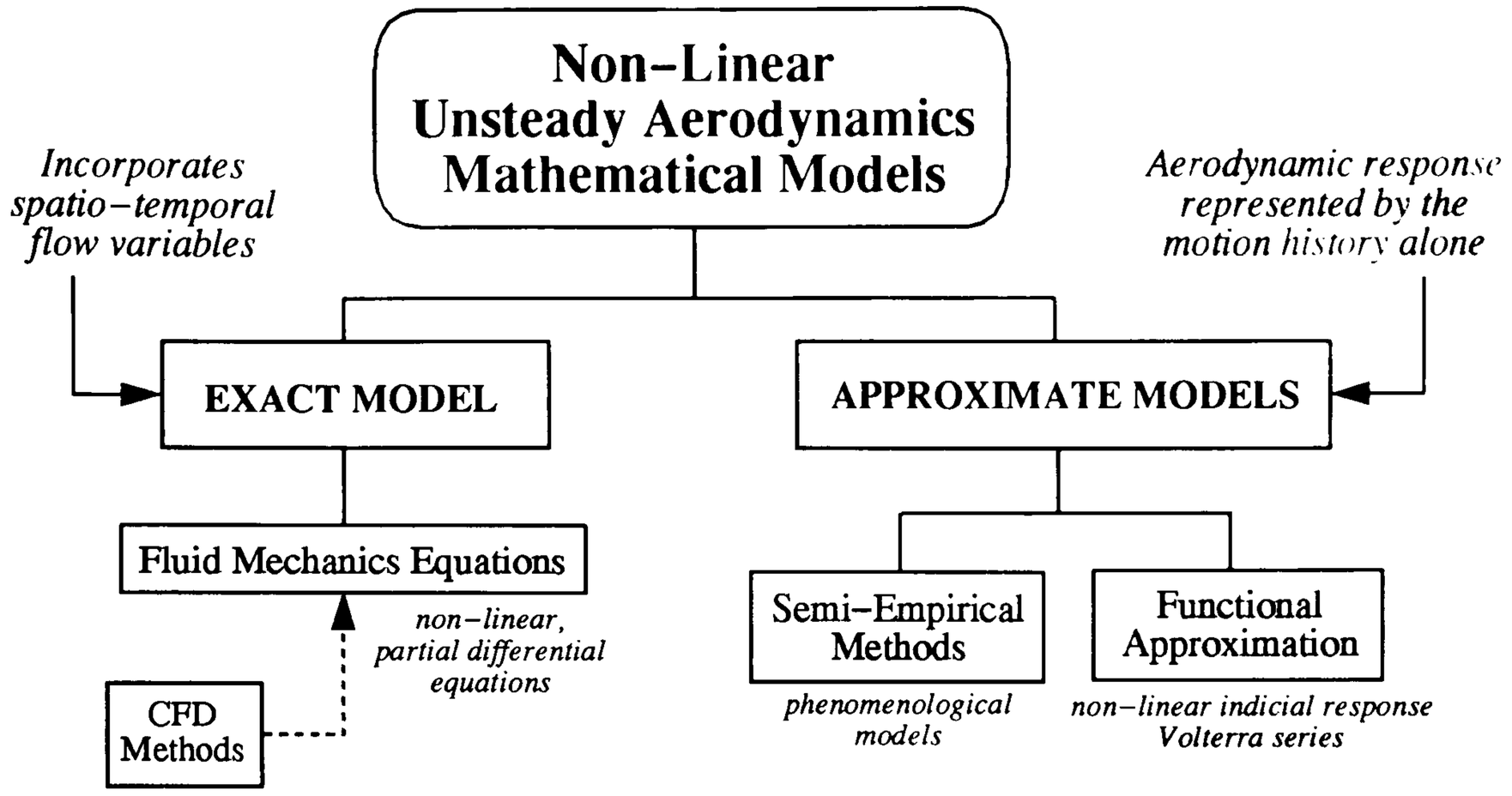


Figure 1.3: Mathematical approaches to non-linear unsteady aerodynamic modelling.

a mathematical and logic formulation of the events that determine the unsteady aerodynamic response over a range of incidence motions and flow regimes. The works by Beddoes [1976, 1982a, 1982b], Tran & Petot [1981], Leishman & Beddoes [1986], and Mahajan *et al.* [1993], are examples of contributions to semi-empirical modelling. The nature of semi-empirical methods facilitates their incorporation into aeroelastic stability and control design. In addition, semi-empirical methods have the advantage of being computationally fast. Nevertheless, semi-empirical models need extensive, specific and precise experimental data. There is also the problem of correlating this data with mathematical and logic formulations.

A summary of the mathematical approaches for unsteady aerodynamic modelling is depicted in Figure 1.3. As functional theory provides a rigorous mathematical framework for non-linear systems modelling, this suggests that a suitable approximation of non-linear unsteady aerodynamic models should consider functional formulations.

1.2 Functional Approximation of Non-Linear Unsteady Aerodynamic Response

The relevance of the functional concept to unsteady aerodynamics is evident in the context of modelling non-linear time-invariant hereditary systems. Functional approximation furnishes an appropriate mathematical framework to model the relationship between unsteady aerodynamic responses and motion history effects.

A coherent modelling approach towards a general non-linear unsteady aerodynamic response functional has been followed by Tobak and co-workers: Tobak & Pearson [1964], Tobak & Schiff [1978, 1981] and Tobak & Chapman [1985]. For example, Tobak assumes the (scalar) unsteady generalised aerodynamic response, $F(t)$, as a functional, $\mathcal{F}$, of the (scalar) generalised motion history, $u_t : u_t(\tau) = u(t+\tau)$, $-\infty < \tau < 0$; that is,

$$F(t) = \mathcal{F}[u_t] \tag{1.6}$$

The unsteady aerodynamic response in Equation 1.6 can be achieved by superposition of indicial response functions (Tobak & Pearson, 1964) to produce a non-linear functional form. The methodology can be generalised by assuming the development of non-linear unsteady aerodynamic functionals in which the indicial function is replaced by a functional. This allows the indicial response to be free from linear assumptions, but still depend on past values of the motion history. Here, the indicial response to a step change is established by the difference between two motion histories, that are identical up to a certain time, when a different step value is applied in each case. Figure 1.4 illustrates the formation of the non-linear indicial response, assuming the case of non-linear unsteady aerodynamic force response $F(t)$ to variations in the generalised motion $u(t)$.

The indicial response is formed by considering two different generalised motions

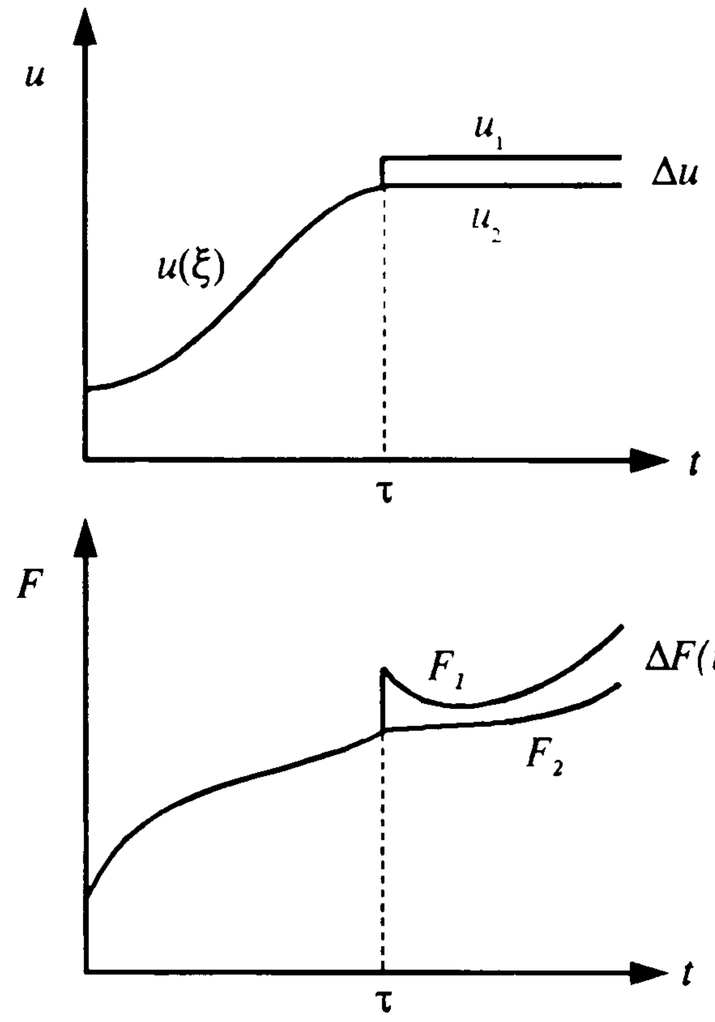


Figure 1.4: Representation of non-linear indicial response formation.

$u(\xi)$ up to time τ, for $0 < \xi < \tau$. At time τ, each case assumes a different motion value, u_1 and u_2, that remains constant for $t > \tau$. The resulting aerodynamic force histories, in each case, differ only for $t > \tau$, and the difference $\Delta F(t)$ (linear or non-linear) can be determined. Then, the *non-linear indicial response* is the limit, for Δu approaching zero, of the ratio $\frac{\Delta F(t,\tau)}{\Delta u}$; that is,

$$\lim_{\Delta u \to 0} \frac{\Delta F}{\Delta u}(t,\tau) = F_u\left[u(\xi); t, \tau\right] \tag{1.7}$$

By applying a generalisation of the superposition principle, integral forms for the aerodynamic force response are achieved. Therefore, by assuming that the non-linear indicial responses must exist and be unique for all values of their arguments ($\xi > \tau$), the resulting non-linear unsteady aerodynamic force at time t is given by the following integral form:

$$F(t) = F(0) + \int_0^t F_u\left[u(\xi); t, \tau\right] \frac{d}{d\tau} u(\tau) d\tau \tag{1.8}$$

The assumption of uniqueness of the indicial responses implies the exclusion of all cases where discontinuities in the functional response occur; for example, unsteady aerodynamic response in separated flowfields. In this case, the discontinuity is characterised by the replacement of an initially unstable state by a new stable equilibrium state, resulting in the non-uniqueness of the indicial response. Tobak & Chapman [1985] present a study on the representation of aerodynamic functionals for discontinuous behaviour.

Detailed mathematical development, leading to Equations 1.7 and 1.8, is described by Tobak & Pearson [1964] and Tobak & Schiff [1981]. The general form of the functional given by Equation 1.8, essentially provides an approximate representation for the non-linear unsteady aerodynamic response; nevertheless, its use is exceedingly complex. In practice, the utilisation of unsteady aerodynamic response functionals depends on further simplifications, as proposed by Tobak & Schiff [1978, 1981] and Jenkins [1991].

Another possibility for modelling non-linear unsteady aerodynamic response in the field of functional approximation, is by functional series methods; for example, *Volterra functional series* (Schetzen, 1980, 1981). This functional form was developed as a generalisation of the Taylor series for a function, and the basic premise of the Volterra series approach is that an exact description for a continuous non-linear time-invariant system; in the context of aerodynamic response, is provided by an infinite series of multi-dimensional convolution integrals of increasing order: that is,

$$F(t) = \sum_{i=1}^{\infty} \underbrace{\int_0^t \cdots \int_0^t}_{i \text{ integrations}} h_i^V(\tau_1, \ldots, \tau_i) \prod_{j=1}^{i} u(t-\tau_j) d\tau_1 \ldots d\tau_i \qquad (1.9)$$

where $F(t)$ is the unsteady aerodynamic force response to the generalised motion $u(t)$ and h_i^V is the i^{th}-order *Volterra kernel*.

The first Volterra kernel represents the linear response of the system to a unit impulse input, while the higher-order kernels are the non-linear responses of the system to multiple (with respect to the kernel order) unit impulse inputs. The higher-order kernels are measures of the non-linearity, or the relative influence of a previous input on the current response, that characterises the temporal effect to the non-linear system. Identification of non-linear systems based on Volterra series requires the determination of the higher-order kernels. This requirement is a major drawback of using this representation. Some approaches overcome this problem by simply assuming that the system is *weakly* non-linear. In this case, the system can be represented with only a few kernels in the Volterra series; for instance, up to the third-order kernel.

For unsteady aerodynamic response prediction, an application of the Volterra series approach has been presented by Silva [1993a, 1993b]. The prediction of general-frequency, non-linear unsteady aerodynamic responses in the transonic regime is carried out by determining a second order Volterra series identified from aerodynamic data provided by an appropriate CFD code. The benefits of the Volterra series approach for subsequent use in *aeroservoelastic* analysis and design in terms of a bilinear systems representation is also presented. In terms of unsteady separated flow models, the applicability of this formulation remains for non-linear attached to weakly separated flows because of the limitations imposed by its restriction to continuous functionals.

Related to an expansion of the Volterra type of functional series, the *Wiener methods* (Billings, 1980 and Schetzen, 1980, 1981) provide other potential identification schemes for non-linear dynamic systems. The first Wiener method considers the idea of representing each functional term by a Fourier-Hermite series. Laguerre functions are used for the Fourier or memory portion of the functional representation, and this is followed by an expansion using normalised Hermite polynomials.

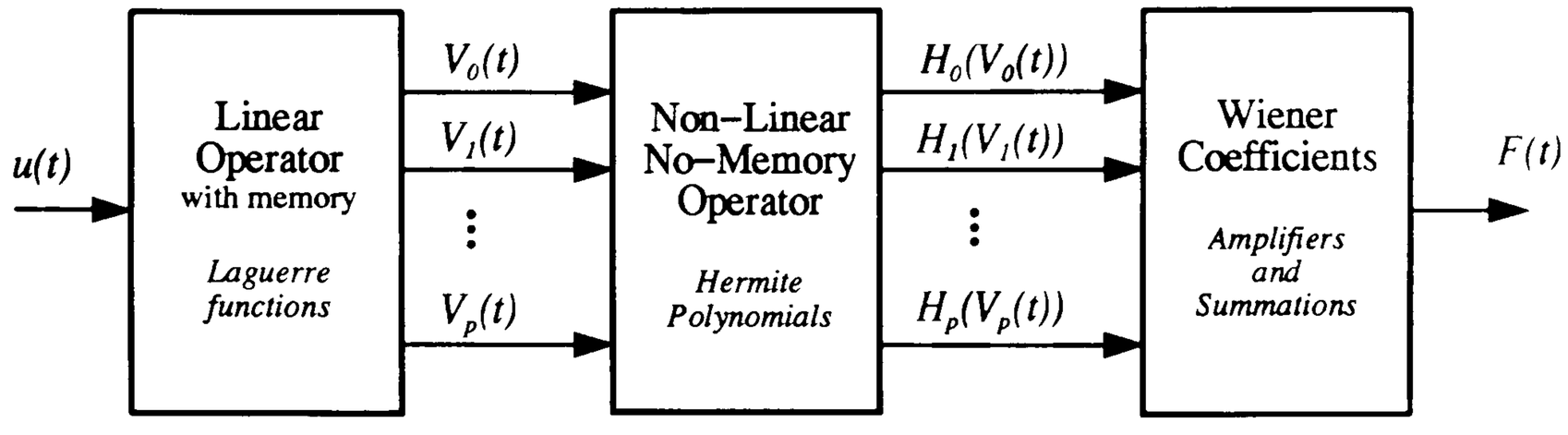


Figure 1.5: Schematic representation of Wiener system.

The synthesis of non-linear systems, in the context of unsteady aerodynamics, using the first Wiener method, as illustrated in Figure 1.5, can be thought of as a cascade process. A linear operator representing the expansion of the past of the generalised motion, $u(t)$, in terms of Laguerre functions has its multiple outputs, $V_i(t)$, transformed by a non-linear no-memory operator based on Hermite polynomials, $H_i(\cdot)$. Then, the outputs of the non-linear operator are amplified by the Wiener coefficients and summation yields the unsteady aerodynamic response, $F(t)$.

The second Wiener method is based on the expansion of a non-linear functional into a series of mutually orthonormal polynomial functionals, the so-called G-functionals. This functional series representation is also equivalent to the first Wiener method series, when white Gaussian inputs are used. Although, the Wiener methods provide a systematic approach to non-linear identification problems, the excessive number of coefficients required to identify the functional series, even for lower-order non-linear systems, makes this approach impractical and difficult to apply.

Other techniques for non-linear dynamic systems identification are based on *block-oriented* approches (Billings & Fakhouri, 1978, 1979, 1982; Billings, 1980; Korenberg & Hunter, 1986 and Hunter & Korenberg, 1986). These approaches represent systems by means of cascade structures of combinations of linear dynamic and non-linear static subsystems. The first Wiener method is in essence a block-oriented

one, as a linear operator with memory is cascaded with a non-linear no-memory operator, as illustrated in Figure 1.5.

The *Hammerstein model* is a block-oriented representation (Billings & Fakhouri, 1979 and Hunter & Korenberg, 1986) of non-linear systems, in which a static non-linearity is followed by a linear dynamic subsystem. Similarly, system models that consist of a cascade of linear dynamic subsystem, a static non-linearity, and another linear dynamic subsystem, or the *LNL systems* (Korenberg & Hunter, 1986), provide another approach in non-linear identification by combining the ideas from Wiener and Hammerstein cascade models.

These techniques have been developed strictly for random processes, in particular for white Gaussian inputs, in order to systematically obtain the parameters of the identified models for the associated class of dynamic systems. These features suggest that the application of block-oriented model realisations via the current methods for the nonlinear unsteady aerodynamic response identification is questionable.

In addition to the difficulties in determining the parameters associated with the aforementioned non-linear functional approximation approaches, other major drawbacks can be associated with the Volterra-Wiener functional series, and block-oriented approaches for non-linear unsteady aerodynamic response modelling. A drawback is that the functional approaches can only be reasonably achieved for single input single output models, due to the increasing complexity of multi-variable functional forms for each respective approach. Appropriate aerodynamic response models should provide values of the generalised forces by means of a single model. Moreover, the inclusion of static inputs; for example, Mach number or Reynolds number, to the modelling scheme may lead to other complications.

Recently, alternative approaches to the functional approximation of non-linear dynamic systems have been proposed by Chen & Chen [1993] and Modha & Hecht-Nielsen [1993]. Based on the universal approximator theorem (Cybenko, 1989 and

Hornik *et al.*, 1989), Modha & Hecht-Nielsen [1993] have developed the so-called *multi-layer functionals*, based on the premise that any time-invariant non-linear system, characterised by continuous functionals, can be approximated by a non-linear superposition of linear affine functionals defined in arbitrary normed spaces. This approach has basically extended the concepts of the approximation theorem for function representation, to obtain a new class of functional series. In the context of non-linear unsteady aerodynamics; for example, multi-layer functional representation of the functional relationship between unsteady aerodynamic force response and generalised motion histories of an airfoil, can be described as,

$$F(t) \approx \mathcal{MF}[\mathbf{u}_t] = \sum_{i=1}^{k} \zeta_i \, \varphi \left(\theta_i + \mathcal{L}_i[\mathbf{u}_t] \right) \tag{1.10}$$

where k is the number of process units, ζ_i and θ_i are real valued constants, φ is a non-constant, bounded, monotone-increasing continuous function, and $\mathcal{L}_i[\mathbf{u}_t]$ denote linear functionals of the generalised motion vector $\mathbf{u}_t$.

For functional approximation representations of non-linear systems, Modha & Hecht-Nielsen [1993] have also established that multi-layer functionals represent a generalisation of the universal approximation theorem. Multi-layer functionals are functional series that also resemble the cascade formulations of block-oriented approaches, in the sense that each process unit in Equation 1.10 represents a cascade of a linear functional and a non-linear operator. However, the simpler formulation of multi-layer functionals, and also the possible composition into layers of process units, is easier to implement than Wiener series. In addition, multi-layer functionals do not present dimensionality restrictions in the model representation. Volterra-Wiener functional series, and block-oriented approaches do not possess the same advantageous property, because in all cases, the dimension of the respective representation; for instance, the number of non-linear kernels in the Volterra series, determines a

specific model. The implicit parallelism and multiple input multiple output model representation capability, are other attractive properties of multi-layer functionals.

Classes of multi-layer functionals can be obtained from specifying different linear functionals $\mathcal{L}_i$ in Equation 1.10. For example, multi-layer feedforward networks are a class of multi-layer functionals, where arbitrary functionals of p-dimensional real valued spaces are represented as weighted superpositions of affine functionals on p-dimensional real valued spaces, modified by a sigmoidal non-linear function. Although multi-layer feedforward networks are appropriated for many cases in non-linear function approximation, a better class of multi-layer functionals for non-linear dynamic systems is necessary. This class can be achieved by assuming $\mathcal{L}_i$ in Equation 1.10, to be defined in arbitrary normed linear spaces (Modha & Hecht-Nielsen, 1993).

1.3 Aerodynamic Functional Realisation via Temporal Neural Networks

Viewing the implementation of a class of multi-layer functionals to represent non-linear unsteady aerodynamic response, a proper linear functional $\mathcal{L}_i$ (*cf.* Equation 1.10) must be adopted. By assuming a basic linear functional in the form of the convolution integral, the multi-layer functional of the non-linear unsteady aerodynamic force response is given by,

$$F(t) \approx \mathcal{MF}[u_t] = \sum_{i=1}^{k} \zeta_i \, \varphi \left(\theta_i + \int_0^t h_i(\lambda) u(t-\lambda) \, d\lambda \right) \tag{1.11}$$

where k, ζ_i, θ_i and φ are as defined in Equation 1.10, and h_i is the unit impulse response of process unit i.

Modha & Hecht-Nielsen [1993] have shown that the multi-layer functional in Equation 1.11 can be realised by a *temporal neural network* (Wan, 1990a, 1990b; Back & Tsoi, 1991 and Back *et al.*, 1994), that allows practical implementations by means of typical neural networks methodologies.

Temporal neural networks represent a generalisation of the conventional neural network concept (Müller & Reinhart, 1990; Hecht-Nielsen, 1990; Hertz *et al.*, 1991 and Haykin, 1994) to account for dynamic behaviour of input to output variables. Temporal neural networks consist of many basic processing units, called *neurons*, joined by connection paths, or *synapses*, modelled by linear, time-invariant, continuous-time filters. Each neuron receives inputs from one or more other neurons, and the sum (or *activation potential*) is transformed by the *activation function* (normally, a non-linear sigmoid function) to yield the neuron output. The arrangement of neurons in a neural network defines its *architecture*. When, the architecture consists of layers of neurons providing outputs in the same directions (information traffic from the input layer to output layer), the network is called a *multi-layer network*. Figure 1.6 schematises an arbitrary multi-layer network architecture and also a generic temporal neuron model. To allow computational implementation, a finite memory to the synaptic filter can be considered. This discrete-time model form is referred to as a *finite impulse response* (FIR) neural network model, where the network connections are comprised of weight vectors.

The determination of a neural network is done by a *learning* process, or *training*. Basically, training processes are algorithms for adjusting the synaptic weights of neural networks. Theories concerned with training processes are still in their infancy. Most of the work on neural networks has used the *back-propagation algorithm* for supervised training (Haykin, 1994). This algorithm produces a sequence of gradient values normally calculated upon desired and network outputs. Those gradient values are used to update the weights of the network during a backward operation along the

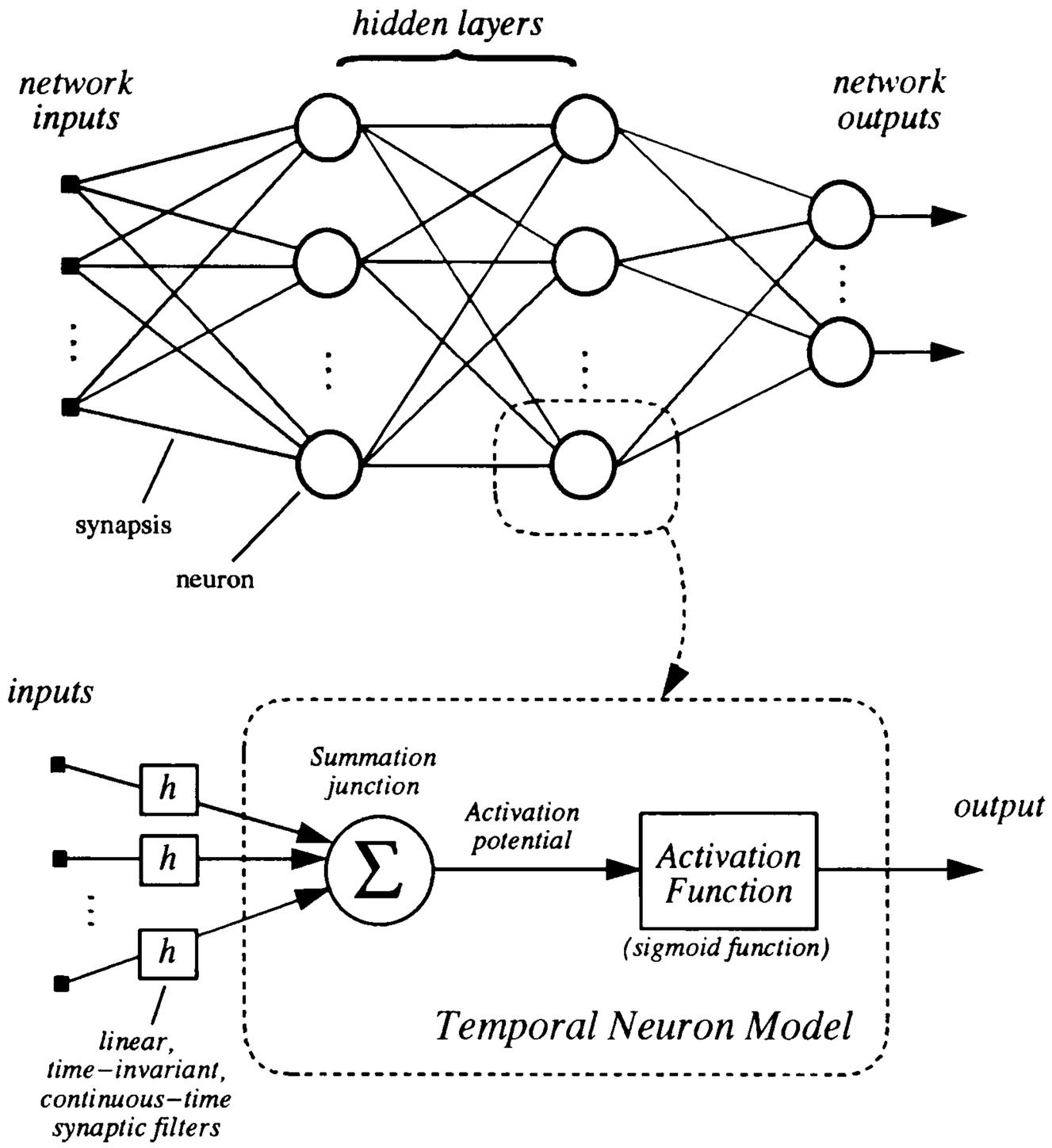


Figure 1.6: Schematic representation of temporal neural network architecture and neuron model.

network connections. Then, the training process becomes a sequence of forward and backward passes through the networks until a point where the difference between desired and network outputs is satisfactorily small. A back-propagation algorithm for FIR neural networks, namely, temporal back-propagation, has been developed by Wan [1990a, 1990b]. In this case, the algorithm works in the same way as in the conventional back-propagation algorithm, but extra information of previous steps in discrete-time of the neurons outputs and activation potentials is required.

Some drawbacks are associated with the temporal back-propagation algorithm. One of them is the need for a differentiable and well-behaved performance index, that may be a limitation for some applications. The back-propagation algorithm

cannot guarantee global optimisation of weights, especially because learning and momentum rates affect the process performance. Moreover, only the same value of time-delay per connection of adjacent hidden layers can be used. Causality restraint problems also compromise the flexibility of the algorithm. Finally, the algorithm can only adapt weight values and as a network's architecture design relies purely on a trial and error basis, the risk of achieving a temporal neural network which overfits the input-output mapping increases. Overfitted networks normally result in bad generalisation property, in other words, a bad identified model.

Studies on techniques to help achieve optimal architecture have not identified a definitive procedure. The use of combinatorial optimisation based on evolutionary programming may be an alternative to the limitations of existing neural network architectures design schemes. Among them, the class of *genetic algorithms* has shown to be a powerful mathematical tool for topology optimisation. Genetic algorithms (Goldberg, 1989; David, 1991; Holland, 1992; Michalewicz, 1992; Beasley *et al.*, 1993a, 1993b; Bäck, 1996 and Mitchell, 1996) are a class of evolutionary algorithm based on combining sequentially structured information of system solutions (*chromosomes*), grouped in a set called the *population*. The information from each possible system solution in the population must be kept in a way to facilitate the reconstruction and evaluation of the system.

Genetic algorithms can be applied to train and adapt temporal neural network architectures, since they are able to combine topological information. Indeed, genetic search allows optimisation of any variables within the same framework, including architecture, learning rules, activation functions, etc. Studies have revealed that genetic algorithms offer an appropriate means to optimise neural network architectures (Fogel *et al.*, 1990; Harp & Samad, 1991; Schaffer *et al.*, 1992; Maniezzo, 1994 and Angeline *et al.*, 1994). Although training neural networks with genetic algorithm does not seem to provide a more efficient scheme to optimise weight val-

ues compared with back-propagation algorithms. genetic algorithms are a promising alternative for training cases where, for example. gradient or error information is not available (Schaffer *et al.*, 1992). For the particular case of FIR neural networks. the application of genetic algorithms for training and adaptation may achieve better results, because of the possibility in assigning different time-delay values per connection, as well as avoiding causality restraint problems.

The wider approximation properties of temporal neural networks (allowed by the multi-layer functional concept) in comparison to conventional neural networks (Poggio & Girosi, 1990 and Narendra & Parthasarathy, 1990), provides a suitable framework for non-linear unsteady aerodynamic response model identification. The systematic way of producing neural network models also offers an attractive opportunity to overcome some of the difficulties related to conventional non-linear system identification approaches (Billings & Fakhouri, 1978, 1979, 1982; Billings, 1980; Korenberg & Hunter, 1986; Hunter & Korenberg, 1986 and Ljung, 1987).

In terms of modelling non-linear unsteady aerodynamic behaviour with neural networks, there are few cases in the recent literature. Specific use of neural networks in modelling unsteady aerodynamics is presented by Faller & Schreck [1995, 1996, 1997] and Schreck *et al.* [1995]. For this case, the authors use a real-time predictive scheme to capture the main features of three-dimensional unsteady separated flowfields. Although important practical results have been achieved, neural networks are basically applied as time-series predictors or function approximators, using a static approach. Further, little has been done to obtain a neural network model which is compatible with the functional representation of unsteady aerodynamic response. The need for an approach that accounts for the functional representation of unsteady aerodynamics is not only a matter of mathematical formalisation, but is essential to accommodate the physical behaviour of non-linear unsteady aerodynamics to a mathematical model.

The present work is concerned with the use of multi-layer functionals in the approximation of non-linear unsteady aerodynamic response in the context of aeroelastic analysis and control. A discrete-time version of the temporal neural network, or finite impulse response (FIR) neural network model, is adopted as a practical realisation of the multi-layer functional model of non-linear unsteady aerodynamic response. A training algorithm, including the optimisation of the FIR neural network architectures, has been developed for the identification of non-linear unsteady aerodynamic response models. The training process is based on a conventional genetic algorithm for the adaptation of the FIR neural network architecture and a simplification of the simulated annealing algorithm (Kirkpatrick *et al.*, 1983; Rutenbar, 1989 and Otten & van Ginneken, 1989) is used to update the weight and bias values of the FIR neural network, and to assist the process in avoiding problems of local minima.

The identification process is used to produce non-linear unsteady aerodynamic response functionals appropriate to variations of two-dimensional airfoil incidence histories for two different flow regimes. In the first case, the weakly non-linear unsteady aerodynamic force response in mildly separated flowfields is examined. The second case considers the non-linear unsteady aerodynamic responses due to compressibility effects in the transonic flow regime. Multiple data sets incorporating boundaries to the incidence motion histories in a range of Mach numbers, are used for the identification process to account for the non-linear behaviour of the unsteady aerodynamic responses.

The approximation properties of the identified multi-layer functionals, in the form of FIR neural networks, are explored by testing the ability to predict the unsteady aerodynamic responses due to incidence motion histories or Mach numbers different from the ones used in the training process. A discussion on the limitations of the multi-layer functional models for unsteady aerodynamic response is also presented.

1.4 Organisation of the Thesis

In this thesis, the application of multi-layer functionals for the approximation of non-linear unsteady aerodynamic response is examined, as well as the advantages and limitations of such an approach. A survey on the main topics related to the difficulties involving the modelling of non-linear unsteady aerodynamic response in the context of aeroelastic analysis and control is introduced in Chapter 1. Practical limitations in the application of the fluid mechanics equations in aeroelastic models have motivated the adoption of approximate methods for modelling unsteady aerodynamic responses. The mathematical approaches based on functional representations have furnished a suitable framework to account for the strong dependence on time-history effects of non-linear unsteady aerodynamic responses. Various forms of functional representations can be used to identify unsteady aerodynamic response models; for example, superposition of non-linear indicial responses and Volterra series. Nevertheless, these functional forms are complicated to implement. In addition, other approaches applied in identification of non-linear dynamic systems; for example, the Wiener methods and block-oriented models, present complications for unsteady aerodynamic modelling. Recently, multi-layer functionals have been introduced as a novel functional series form that can be realised via temporal neural networks. The application of multi-layer functionals for the approximation of the unsteady aerodynamic response is proposed.

In Chapter 2, the theoretical foundations of multi-layer functionals are presented. Initially, the basic issues on approximation theory are introduced in the context of neural networks. For non-linear dynamic systems, neural networks have been heuristically used as parametric input-output models. The success of these models, in relation to the traditional non-parametric and block-oriented non-linear sytem identification approaches, has motivated studies on rigorous mathematical

formulations to justify the neural network approximation properties. The result is the universal approximation theorem, establishing that any continuous function can be approximated by linear finite combinations. Then, based on this foundation, generalisations to the formulation have been proceeded, in order to adequate the theorem to account for functional representations. This leads to the definition of a novel parametric family of real valued mappings, named multi-layer functionals. Then, the practical realisation of multi-layer functionals in terms of temporal neural networks in discrete-time, or the finite impulse response (FIR) neural networks, to represent time-invariant sytems are presented.

The genetic search used in the supervised training of the FIR neural network for the identification of unsteady aerodynamic response models, is presented in Chapter 3. Initially, theoretical aspects of genetic algorithms are overviewed. A detailed explanation on how the FIR neural networks are encoded in chromosomes is presented. The genetic operators, applied in the optimisation of FIR neural network architectures, are then described. Then, the use of simplifications of the simulated annealing algorithm, applied to update the weight and bias values and to assist the process to avoid stagnation on the convergence performance, are explained.

Aspects of multi-layer functional representation of non-linear unsteady aerodynamic models are presented in Chapter 4. The identification of unsteady aerodynamic response multi-layer functionals, in the form of FIR neural networks, is carried out for two different flow regimes. The first case considers the weakly non-linear unsteady aerodynamic force response to variations in the incidence motion histories in mildly separated flowfields for a range of Mach numbers. A semi-empirical model is used to generate the necessary two-dimensional aerodynamic data for the identification process. In the second case, flow regimes influenced by compressibility effects are considered. The functional of the non-linear unsteady aerodynamic responses to variations of the incidence motion histories of a two-dimensional NACA 0012 airfoil

is identified, using aerodynamic data from a CFD code based on the solution of the Euler equations. A discussion on the algorithm performance and on the approximation properties of the identified models are also presented. Limitations to the multi-layer functional approach are identified.

Finally, in Chapter 5, the conclusions and directions for future investigations are presented.

Chapter 2

Multi-Layer Functional Models of Non-Linear Dynamic Systems

2.1 Introduction

Identification of approximate models of dynamic systems remains a common problem, particularly when systems present complex non-linear behaviour and a strong dependence on the past history of the input variables. Approaches for non-linear system identification can be categorised in (Korenberg & Hunter, 1986): *(i)* kernel or non-parametric methods; *(ii)* block-oriented models; and *(iii)* parametric approaches. In the first category, the Volterra and Wiener functional series are among the most important ones. The block-oriented models represent a larger category, where cascades of interconnected linear dynamic systems and non-linear static operators are assumed. Surveys on these approaches are presented in Billings [1980], Billings & Fakhouri [1982], Korenberg & Hunter [1986] and Hunter & Korenberg [1986].

Parametric models also provide suitable alternatives for non-linear system identification (Billings, 1980). Recently, the use of *neural networks* (Haykin, 1994) for non-linear system parametric modelling has increased considerably. Neural networks

consist of superpositions and combinations of parametric linear operators (summations of weighted inputs) in cascade to non-linear operators (activation functions). However, the success of neural networks in the context of system identification is based, historically, on heuristic approaches rather than on theoretical foundations.

The formalisation and proper mathematical explanation of neural network identification properties for non-linear systems have been achieved by the *universal approximation theorem* (Cybenko, 1989 and Hornik *et al.*, 1989). By defining a rigorous mathematical formulation based on the Stone-Weierstrass and Kolmogorov theorems, the universal approximation theorem establishes that finite combinations of superpositions of a fixed, univariate function over a set of affine functions can uniformly approximate any continuous multivariable function (Cybenko, 1989). Although the universal approximation theorem has provided a suitable conceptual basis for using conventional neural networks as parametric input-output models of non-linear dynamic systems, limitations on the set of internal network parameters requires special architecture modifications.

While many non-linear dynamic systems can be identified by *functions*, the majority of systems are better represented by *functional* forms. Recognising the importance of functional representations in non-linear dynamic systems identification. Chen & Chen [1993] and Modha & Hecht-Nielsen [1993] have studied generalisations of the universal approximation theorem. Although both approaches have provided important developments towards a functional form for the universal approximation theorem, the approach by Modha & Hecht-Nielsen [1993] embraces a more complete representation by establishing a functional form that can uniformly approximate any continuous functional on a normed linear space.

Modha & Hecht-Nielsen [1993] have produced a novel parametric family of real valued mappings on arbitrary normed linear spaces, named *multi-layer functionals*. Multi-layer functionals can also be used to represent time-invariant, continuous-time

or discrete-time, finite or infinite memory, causal or non-causal systems. Moreover, the resulting class of functional forms for parametric non-linear systems representation can also be realised by means of temporal neural networks (Wan, 1990a, 1990b; Back & Tsoi, 1991; Back *et al.*, 1994 and Haykin, 1994)

The purpose of this chapter is to present the theoretical foundations of multi-layer functionals and also to demonstrate how multi-layer functionals can be used as parametric input-output models of non-linear dynamic systems. Primarily, the universal approximation theorem is described in the context of neural networks for non-linear system identification. The development of functional forms by expanding the concepts involved in the universal approximation theorem is presented. In this context, the approaches by Chen & Chen [1993] and Modha & Hecht-Nielsen [1993] are described and the definition of multi-layer functionals is presented. Then, multi-layer functionals are defined to facilitate the modelling of time-invariant, continuous-time, infinite memory, and anti-causal dynamic system. The resulting formulation can be shown to be realised by means of *temporal neural networks* (Wan, 1990a, 1990b; Back & Tsoi, 1991; Back *et al.*, 1994 and Haykin, 1994).

Finally, temporal neural networks are described and the finite impulse response (FIR) neural model is described as a practical computational implementation of temporal neural networks in discrete-time to represent, time-invariant, finite memory, and causal dynamic systems. Using this approach, conventional use of neural networks methodologies in the production of parametric input-output models of non-linear dynamic systems is facilitated.

2.2 Multi-Layer Functional Approximation

The principles of the universal approximation theorem form a proper basis for function approximation. This approach has provided suitable framework for non-linear dynamic systems identification and modelling; for example, by using neural networks. However, better dynamic system representation is achieved by using functional forms. Generalisations to the universal approximation theorem has been produced, and the result are new functional forms that furnish appropriate framework for parametric input-output models of non-linear dynamic systems. The mathematical foundations to the definition for a class of functionals, called multi-layer functionals, are presented in this Section.

2.2.1 Issues on Function Approximation

The approximation of mathematical functions or mappings represents a central problem in a variety of subjects. The basic issues of approximation theory, that are relevant to the formulation of the *universal approximation theorem*, are presented in this section. Although this theorem has been developed to explain the approximation properties of neural networks, its significance to approximation theory is more far-reaching (Haykin, 1994).

The methods based on superposition of continuous functions have established the foundations of the universal approximation theorem. The basis of this theorem comes from the Stone-Weierstrass and Kolmogorov theorems (Cybenko, 1989; Hornik *et al.*, 1989 and Hecht-Nielsen, 1990), viewing the systematisation and classification of theoretical foundations of *neural networks* for function approximation. The Stone-Weierstrass theorem presents a simple criterion to define functions used to uniformly approximate arbitrary continuous functions, while the Kolmogorov theorem establishes a superposition formulation for the function approximation problem

(Hecht-Nielsen, 1990).

The Kolmogorov theorem also represents an important mathematical tool for the particular case of neural networks (Hecht-Nielsen, 1990 and Haykin, 1994). Reformulations of this theorem, in the context of neural networks theory, have provided appropriate mathematical foundations to explain the function approximation properties of neural networks (Hecht-Nielsen, 1990).

In the context of approximation of functions by neural networks, the approaches using finite linear combination forms have been rigorously represented by the *universal approximation theorem* (Cybenko, 1989 and Hornik *et al.*, 1989):

> *If φ is a non-constant, bounded, and monotone-increasing continuous function, then, any function y belonging to the space of continuous functions on the p-dimensional unit hypercube $I \equiv [0,1]$ can uniformly be approximated by a finite linear combination of k process units such that,*
>
> $$\left| y(x_1, x_2, \ldots, x_p) - \sum_{j=1}^{k} \zeta_j \varphi \left(\theta_j + \sum_{i=1}^{p} w_{ji} x_i \right) \right| < \varepsilon \qquad (2.1)$$
>
> *where $\zeta_j, \theta_j, w_{ji} \in \Re$, $\varepsilon \in \Re_+$, $\{x_1, x_2, \ldots, x_p\} \in I$, and $|\cdot|$ represents an appropriate norm operator.*

2.2.2 Functional Approximation

Approximation of functions has proved useful in applications to system modelling, identification and realisation (Narendra & Parthasarathy, 1990 and Masri *et al.*, 1993). Although approximation approaches based on Inequality 2.1 are concerned with continuous functions, in practice dynamic system models can be viewed as

functionals defined on some set of functions (Chen & Chen, 1993). This issue has motivated Chen & Chen [1993] and Modha & Hecht-Nielsen [1993], in the development of a generalisation for the theorem described in Inequality 2.1.

A functional is a mathematical relationship between sets of points, being its domain sets of functions. For a set of real valued functions of t, the functional $\mathcal{F}$ assigns a real number $Y(t)$ to a real valued function x_t from its domain; that is,

$$Y(t) = \mathcal{F}[x_t] \tag{2.2}$$

Indeed, the number $Y(t)$ depends on all ordinates x_t that correspond to abscissas t of a previously defined time interval. A functional can be considered as a function with an entire continuum of independent variables, which permits functionals to be considered as a generalisation of the mathematical concept of function.

By approximating a generic functional $\mathcal{F}[x_t]$ by a function of $m-1$ variables given by $Y(g(x_1), \ldots, g(x_{m-1}))$, where $g(\cdot)$ are generic functions, Chen & Chen [1993] have achieved the following formulation, that generalises the universal approximation theorem in Inequality 2.1; that is,

> *If U is a set in a space of infinite dimension $C[p_1, p_2]$, $\mathcal{F}$ is a continuous functional defined on U, and $\varphi(\cdot)$ is a non-constant, bounded, and monotone-increasing continuous function, then for any $\varepsilon \in \Re_+$, there exist $m+1$ points $p_1 = t_0 < \ldots < t_m = p_2$, a positive integer k and $\zeta_j, \theta_j, w_{ji} \in \Re$, such that*
>
> $$\left| \mathcal{F}[x_t] - \sum_{j=1}^{k} \zeta_j \varphi \left(\theta_j + \sum_{i=0}^{m} w_{ji} x_{t_i} \right) \right| < \varepsilon. \quad \forall x_t \in U \tag{2.3}$$
>
> *where $|\cdot|$ represents an appropriate norm operator.*

The significance of Inequality 2.3 for dynamic system representations is the fact that the formulation provides a theoretical basis for approximation of continuous functionals.

Simultaneuously, Modha & Hecht-Nielsen [1993] have also used the functional approach to establish generalised formulations to the universal approximation theorem. The Modha & Hecht-Nielsen [1993] approach presents the same structure of Inequality 2.3, differing in the way the argument of φ is defined. This argument is assumed to be an affine functional $\mathcal{A}[x_t]$ of the form:

$$\mathcal{A}[x_t] = \theta + \mathcal{L}[x_t] \tag{2.4}$$

where $\theta \in \Re$, $\mathcal{L}[x_t]$ belongs to the set of all continuous linear functionals on arbitrary normed linear spaces.

Although the differences between both approaches seem to be only a matter of notation, the Modha & Hecht-Nielsen [1993] approach is more far-reaching, because no approximation to the affine functionals is assumed to be necessary.

2.2.3 Multi-Layer Functionals

Modha & Hecht-Nielsen [1993] have expanded the concept of the universal approximation theorem (*cf.* Inequality 2.1), by adopting a generalisation of the affine form in Equation 2.4. For that, a linear functional on an arbitrary normed linear space has been defined as,

$$\mathcal{L}[x_\eta] = \int_I x(\eta) d\mu(\eta) \tag{2.5}$$

where $\eta \in I \equiv [0, 1]$, x_η belongs to the set of real valued continuous functions with domain I, and μ belongs to the set of functions of bounded variation on the unit

hypercube I.

When the linear functional in Equation 2.5 is used in Equation 2.4, the formulation reveals a class of functional forms defined over a set of all real valued, continuous functions with domain $I \equiv [0, 1]$. This functional class has been named by Modha & Hecht-Nielsen [1993], as *multi-layer functionals*; that is,

$$\mathcal{MF}[x_\eta] = \sum_{j=1}^{k} \zeta_j \varphi \left(\theta_j + \int_I x(\eta) d\mu_j(\eta) \right) \tag{2.6}$$

where ζ_j, $\theta_j \in \Re$, $x(\eta)$ and $\mu_j(\eta)$ are defined by Equation 2.5, and k is the number of process units.

To use multi-layer functionals as input-output representations for dynamic systems, the notion of time needs to be incorporated. A general formulation for time-invariant, continuous-time, infinite memory, and anti-causal system representations by multi-layer functionals is presented by Modha & Hecht-Nielsen [1993]. A basic linear functional $\mathcal{L}[x_t]$ in the convolution form is assumed; that is,

$$\mathcal{L}[x_t] = \int_0^t h(\lambda) x(t - \lambda) d\lambda \tag{2.7}$$

where h is the unit impulse response due to $x(t)$.

Then, the application of the linear functional in Equation 2.7, to define an affine functional of the form of Equation 2.4, results in the following multi-layer functional, which approximates the dynamic system response $y(t)$:

$$y(t) \approx \mathcal{MF}[x_t] = \sum_{j=1}^{k} \zeta_j \varphi \left(\theta_j + \int_0^t h_j(\lambda) x(t - \lambda) d\lambda \right) \tag{2.8}$$

where ζ_j, $\theta_j \in \Re$, φ is a bounded, continuous function, and h_j is the unit impulse response of process unit j, for $j = 1, \ldots, k$.

2.3 Neural Network Realisation of Multi-Layer Functionals

A close examination of Equation 2.8 reveals that it corresponds to the definition of *temporal neural networks* described by Wan [1990a, 1990b], Back & Tsoi [1991], Back *et al.* [1994], and Haykin [1994]. This allows a suitable framework for practical representation of a large class of non-linear dynamic systems.

2.3.1 Temporal Neural Networks

Temporal neural networks as described in this Section, comprise the category of neural networks represented by a spatio-temporal neuron model joined by connecting links called synapses (Wan, 1990a, 1990b; Back & Tsoi, 1991; Back *et al.*, 1994 and Haykin, 1994). In this case, the synapses are modelled by *linear, time-invariant, continuous-time filters.*

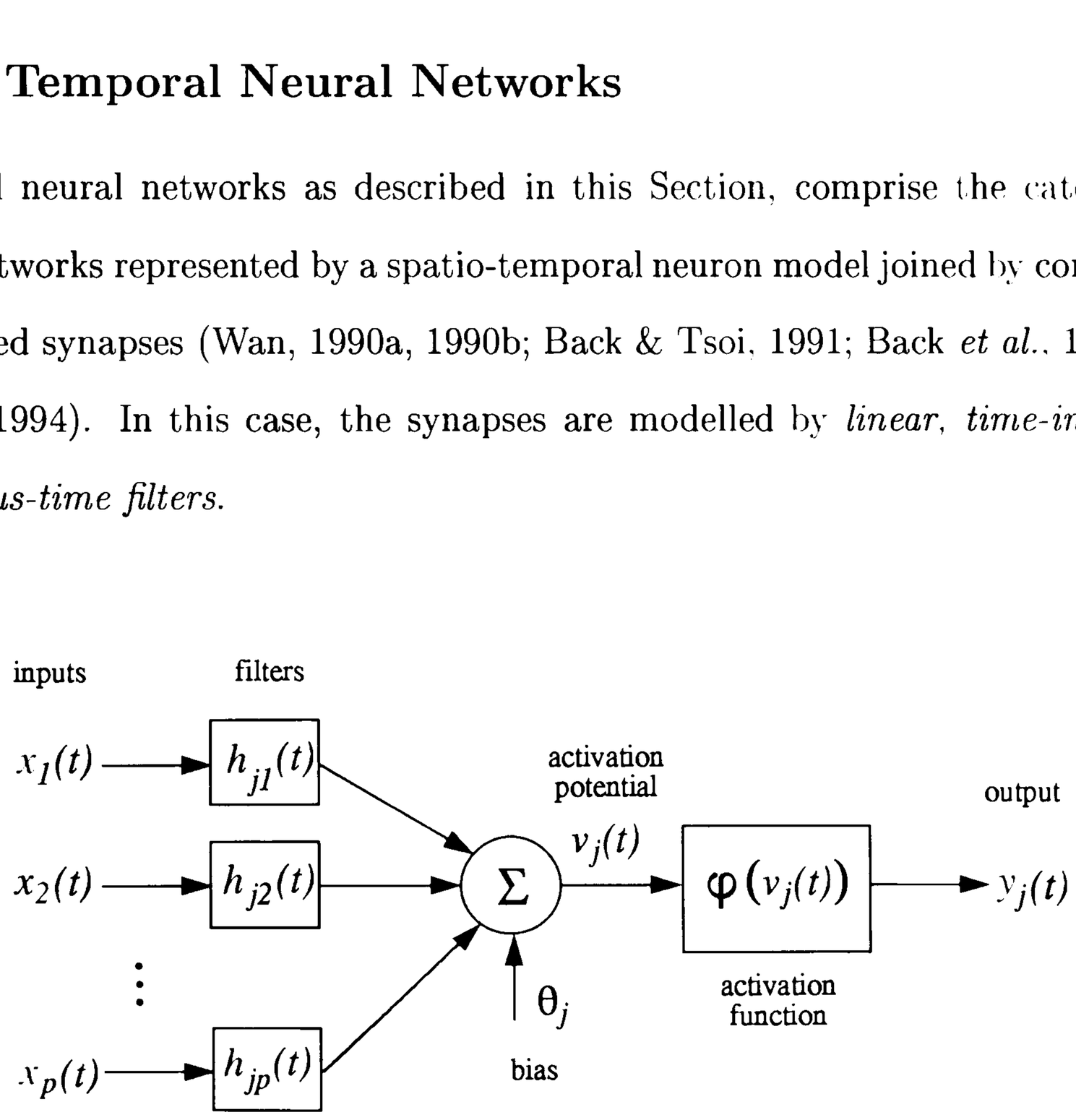


Figure 2.1: Temporal neuron model.

Figure 2.1 illustrates the neuron model, in which the synapse i belonging to the neuron j has its temporal behaviour described by an impulse response $h_{ji}(t)$. For

the input $x_i(t)$ denoting the excitation applied to synapse i (for $i = 1, \ldots, p$), the synaptic response is determined by the convolution of the impulse response $h_{ji}(t)$ with $x_i(t)$. For a neuron j with a total of p synapses, the associated *activation potential* $v_j(t)$ is due to the combined effect of all the inputs and the applied bias values θ_j is given by,

$$v_j(t) = \theta_j + \sum_{i=1}^{p} \int_0^t h_{ji}(\lambda) x_i(t - \lambda) d\lambda \tag{2.9}$$

The neuron output $y_j(t)$ is obtained by applying the *activation function* φ, normally a sigmoidal function, on $v_j(t)$; that is,

$$y_j(t) = \varphi(v_j(t)) \tag{2.10}$$

An illustration of a simplified representation for the temporal neural model is shown for the neuron j in Figure 2.2.

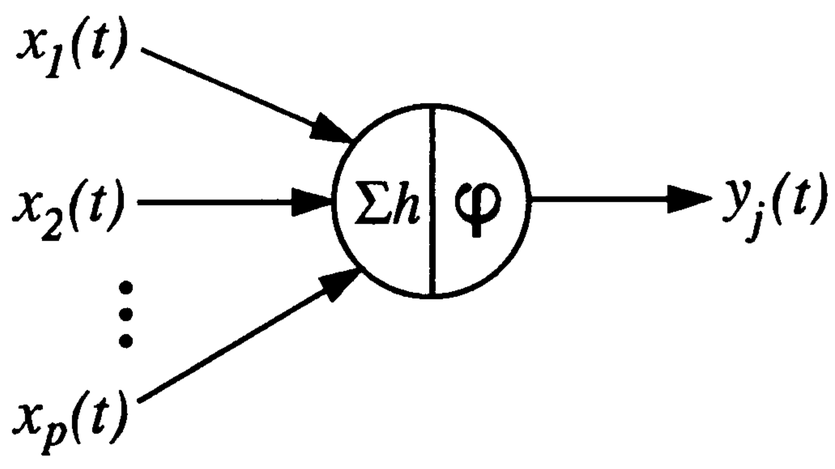


Figure 2.2: Simplified representation for a temporal neuron.

A multi-layer temporal neural network is formed by composing layers of neurons. A schematic representation (using the simplified neuron representation shown in Figure 2.2) of a multi-layer network architecture, for the input-output pair $(x(t), y(t))$,

composed of neurons modelled by Equations 2.9 and 2.10, distributed in two hidden layers, is depicted in Figure 2.3.

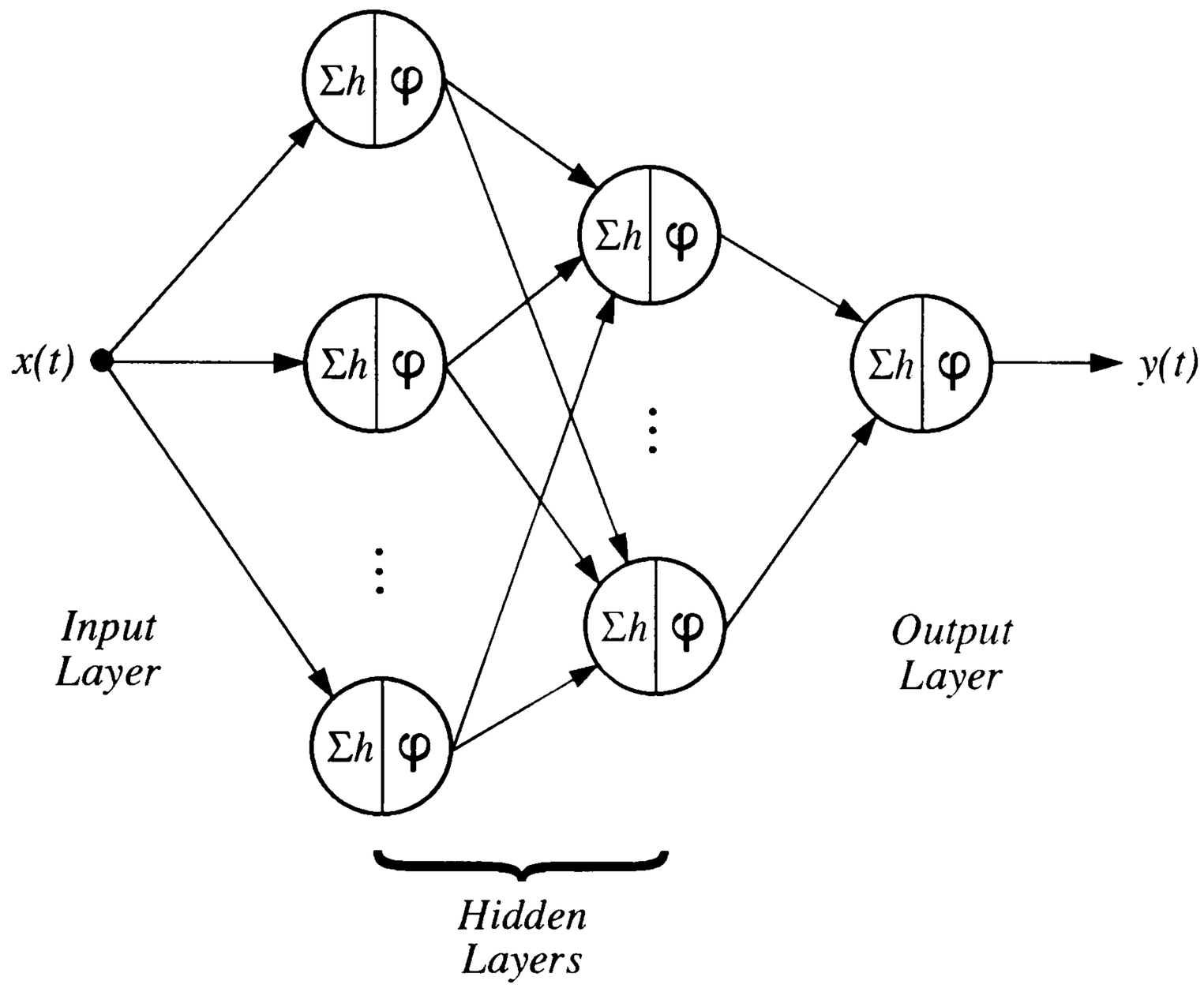


Figure 2.3: Temporal network architecture.

The temporal neuron model, described by Equations 2.9 and 2.10, is consistent with the representation of multi-layer functionals given by Equation 2.8. The representation in Equation 2.8 can be interpreted as a special form of multi-layer neural network possessing a *single* hidden layer of temporal neurons with the network output being the linear combination of each hidden neuron output. In practice, a more robust form of multi-layer functional utilises a network architecture with *multiple* hidden layers. From a computational point of view, it is convenient to assign a discrete-time version of the temporal neuron model, by assuming the synapses as *finite impulse response FIR filters*.

2.3.2 Finite Impulse Response (FIR) Neural Model

To determine the FIR model it is convenient that the continuous-time synaptic model (in fact, a *synaptic filter*) obeys the following characteristics:

- The synaptic filter must be *causal*, that is,

$$h_{ji}(t) = 0, \quad t < 0$$

- The synaptic filter must have *finite memory*, that is.

$$h_{ji}(t) = 0, \quad t > T_{ji}$$

where T_{ji} denotes the *memory span* of the synapse i belonging to the neuron j.

Therefore, Equation 2.9 can be re-written as,

$$v_j(t) = \theta_j + \sum_{i=1}^{p} \int_0^{T_{ji}} h_{ji}(\lambda) x_i(t-\lambda) d\lambda \tag{2.11}$$

For convenience, the convolution integral in Equation 2.11 is approximated by a *convolution sum*, thereby permiting a discrete-time representation. Consequently, the continuous-time variable t is substituted by a discrete-time variable $n\Delta t$, where n is an integer and Δt is the sample interval. Then, Equation 2.11 is approximated as

$$v_j(n) = \theta_j + \sum_{i=1}^{p} \sum_{\ell=0}^{\tau_{ji}} w_{ji}(\ell\Delta t) x_i(\Delta t(n-\ell)) \tag{2.12}$$

where $\tau_{ji} = \frac{T_{ji}}{\Delta t}$ is the number of delay units of the filter in synapse i belonging to the neuron j, and $w_{ji}(\ell\Delta t) = h_{ji}(\ell\Delta t)\Delta t$.

The sample interval Δt has a common uniform value to all the time-varying quantities in Equation 2.12. For notational convenience, Δt may be omitted from

the arguments of all time-varying quantities. Each connection may also present different time-delay values. Therefore, Equation 2.12 can be re-written in the form.

$$v_j(n) = \theta_j + \sum_{i=1}^{p} \sum_{\ell=0}^{\tau_{ji}} w_{ji}(\ell) x_i(n-\ell) \tag{2.13}$$

Equation 2.13 describes the expression for the activation potential of the *finite impulse response (FIR) model*, and the neuron can be illustrated as in Figure 2.4, where $\mathbf{w}_{ji}$ denotes the weight vector ($\Re^{\tau_{ji}+1} \times 1$) of synapse i belonging to neuron j; that is,

$$\mathbf{w}_{ji} = [w_{ji}(0) \quad w_{ji}(1) \quad \ldots \quad w_{ji}(\tau_{ji})]^{\mathrm{T}}$$

and

$$\mathbf{x}_i(n) = [x_i(n-0) \quad x_i(n-1) \quad \ldots \quad x_i(n-\tau_{ji})]^{\mathrm{T}}$$

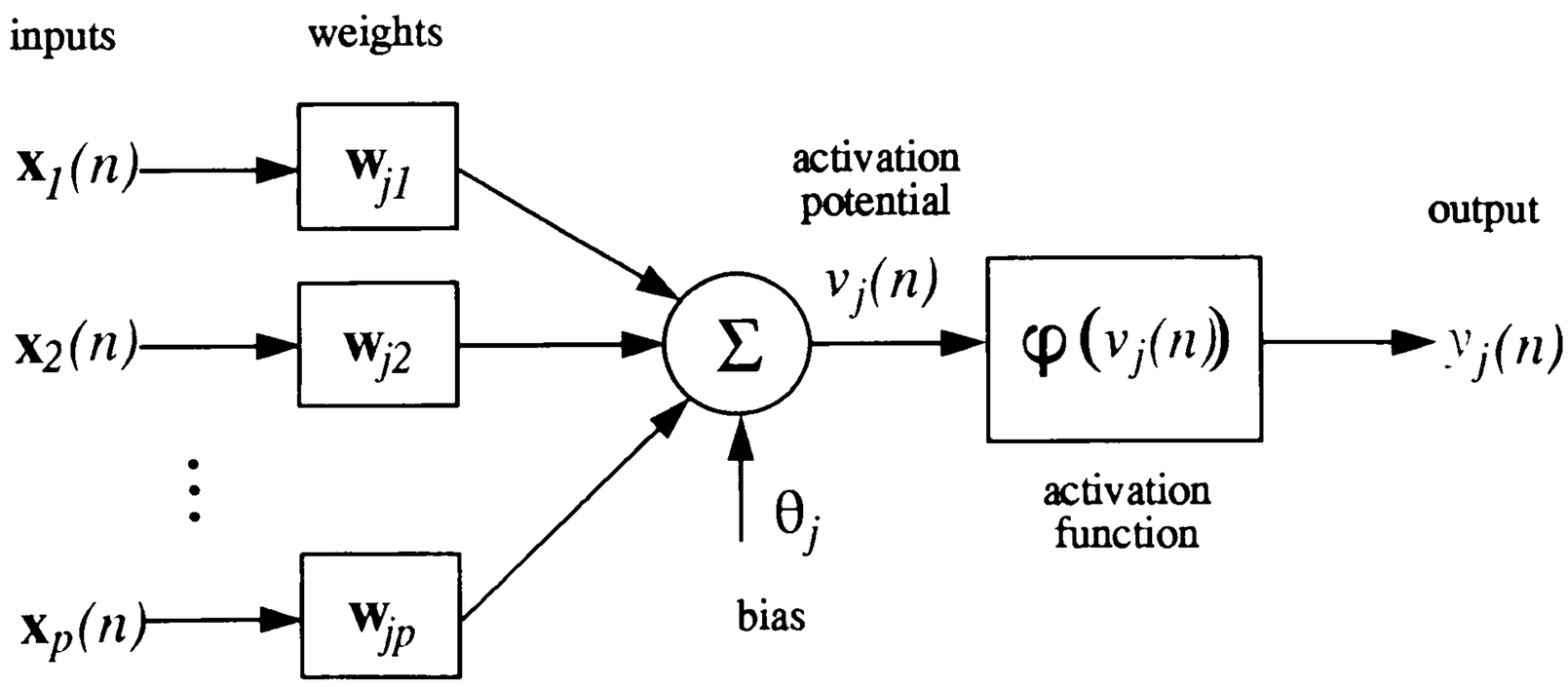


Figure 2.4: FIR neuron model.

A simplified representation for the FIR neural model (neuron j) is depicted in Figure 2.5.

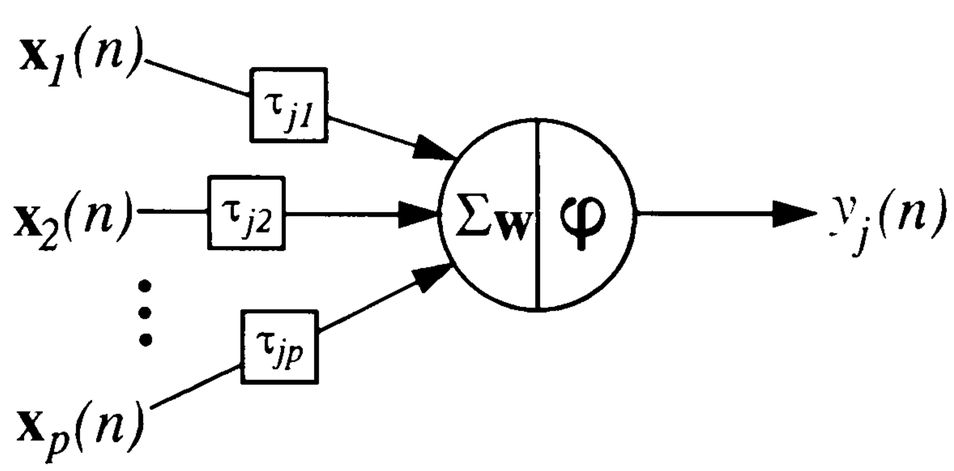


Figure 2.5: Simplified representation for a FIR neuron.

An example of the final representation of a multi-layer FIR neural network architecture is illustrated in Figure 2.6, where each neuron is represented as in Figure 2.5. Figure 2.6 presents an arbitrary FIR neural network with p inputs, k outputs and two hidden layers. The first hidden layer has m neurons, while the second one has q neurons. The neurons of the FIR neural network are enumerated in sequence from the input layer to the output layer, starting in the first hidden layer. Time-delays per connection; that is, τ_{ji} in the connection between neuron i to neuron j, are also depicted. The FIR model is the practical realisation of the discrete-time multi-layer functional used to represent time-invariant, finite memory, and causal systems.

2.4 Summary

The foundations of multi-layer functionals for non-linear dynamic systems representation are described in this chapter. Multi-layer functionals are a new parametric family of real valued mappings compounded by non-linear combination of linear

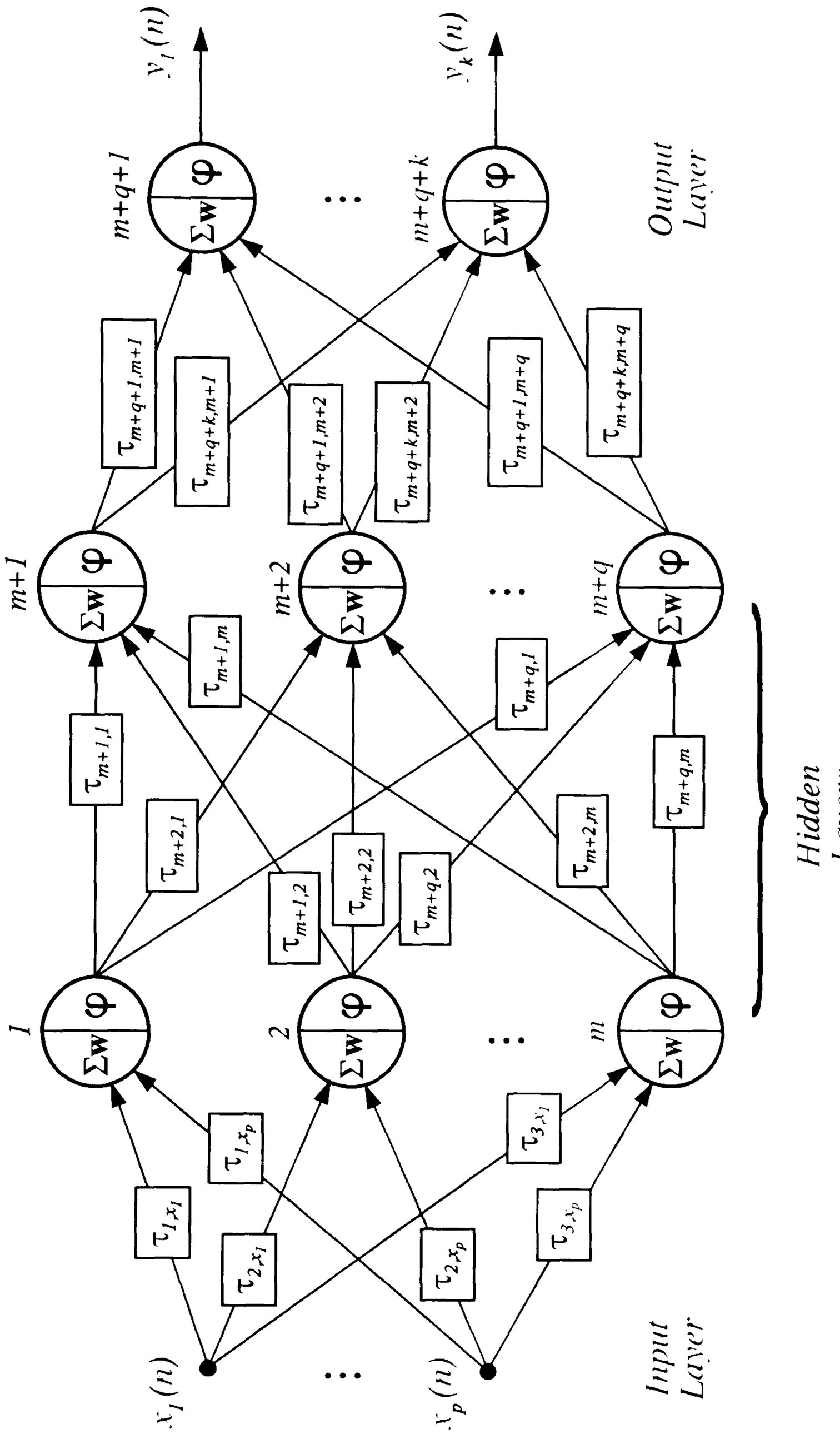


Figure 2.6: FIR neural network architecture.

affine functionals on arbitrary normed linear spaces. Using the concepts of the universal approximation theorem expanded by functional forms, multi-layer functionals provide an appropriate framework for a large class of non-linear system models.

The development of a multi-layer functional to model time-invariant, continuous-time, infinite memory, causal system has been shown to be realisable in terms of temporal neural networks. Moreover, for practical computational implementations, temporal networks in discrete-time, or finite impulse response neural models, can be used.

Chapter 3

Identification of Non-Linear Dynamic Systems using FIR Neural Networks: A Genetic Algorithm Approach

3.1 Introduction

A suitable framework for the representation of non-linear dynamic systems, as shown in Chapter 2, is provided by multi-layer functional models. A practical realisation of the multi-layer functional representation of time-invariant systems can be achieved with a discrete-time, finite memory temporal neural network; that is, the finite impulse response (FIR) neural network. Identification of an appropriate FIR network model is achieved by means of a supervised training process in which both the network parameters and network architecture are adapted to minimise the error between the prescribed outputs and the network outputs.

The back-propagation algorithm (Hertz *et al.*, 1991 and Haykin, 1994) has been, so far, the most important technique in training neural networks. Finite impulse response (FIR) neural network may also be trained with the back-propagation algorithm. In this case, the temporal back-propagation algorithm, as proposed by

Wan [1990a, 1990b], represents the most appropriate algorithm. However, the implementation of the temporal back-propagation algorithm suffers from a number of drawbacks:

(i) The training needs a fixed architecture;

(ii) As back-propagation is based on gradients to update weights, then the cost function has to be well-behaved;

(iii) Because of the filtering, a causal restraint has to be observed, and extra storage of each neuron output at previous time steps is required;

(iv) The time-delays per connection must be fixed with the same value in all connections between adjacent layers.

Since there is not a definitive approach available for designing neural network architectures, the risk of adopting trail-and-error schemes may lead to overfitted approximations and, consequently, bad generalisation properties. Moreover, back-propagation cannot guarantee a global minimum of the cost function, because of the dependence on the training parameters, and on the properties of the input-output mapping.

The restrictions on the internal time-delay distribution may also influence the generalisation features of the FIR network; therefore, a more flexible and systematic scheme to design FIR network architectures is desired. The rigid structure of the temporal back-propagation algorithm also justifies the adoption of alternative methods for training FIR networks.

There is a clear potential in the use of *genetic algorithms* to evolve neural network architectures (Fogel *et al.*, 1990; Harp & Samad, 1991; Maniezzo, 1994 and Angeline *et al.*, 1994), as well as training the internal parameters. Genetic algorithms (Goldberg, 1989, David, 1991 and Holland, 1992) are search algorithms

based on the principles of natural evolution to manipulate and assess the optimum solution among a set of possible solutions (*population*) to a given problem. Basically, each possible solution (*individual*) is encoded into a numerical or symbolic structure (*chromosome*) that is used in specific combinatorial operations (*selection*, *crossover*, *mutation*) to generate a new set of possible solutions.

Recently, the combination of genetic algorithms and neural networks has attracted a wide research interest. The motivation comes from the need for a systematic design of neural network architectures, and other training procedures. Schaffer *et al.* [1992] have surveyed the recent interest in this area and concluded that the greatest advantage of such a combination lies in the use of genetic algorithms to evolve neural network architectures. As far as training neural networks is concerned, genetic algorithms have not produced substantial improvements to the back-propagation algorithm, but genetic algorithms may be a promising training method when gradient or error information is not available. Moreover, there is a need for more investigations on genetic algorithms to train dynamic versions of neural networks (Schaffer *et al.*, 1992).

For the specific combination with FIR neural networks, genetic algorithms can be considered for both architecture adaptation and training. The advantages of architecture adaptation by genetic algorithms are evident. In terms of training FIR neural networks, genetic algorithms seem to be appropriate, since temporal back-propagation presents limitations in dealing with different time-delays per connection and is also subject to causality restraints. However, as genetic algorithms can only encode finite sequences of possible solutions, weight values can be restricted to a limited region of the weight space. This suggests that extra mechanisms for updating weight values should be applied to assist the overall process.

This chapter presents an algorithm to train FIR neural networks and to adapt the network architectures, for the purpose of identification of non-linear dynamic

system models. A genetic algorithm is developed to optimise the number of neurons and the time-delay distribution of FIR neural networks, while a variation of the simulated annealing algorithm is used to update weight and bias values, and to assist the process of avoiding local minima. The chapter continues with an overview on genetic algorithms, where their basic structure and premises are exposed. Then, the procedure to encode FIR neural network architectures, the determination of the fitness function, and the genetic operators for the identification process are presented. Finally, details of the training process are described.

3.2 Overview of Genetic Algorithms

Genetic algorithms are a type of evolution based search algorithms that manipulate sets of possible encoded solutions to a given problem in order to determine the optimum one. As an evolution based technique or evolutionary algorithm, genetic algorithms operate on the set of decoded solutions according to the principles of natural selection and the *survival of the fittest* premise. Detailed descriptions of genetic algorithms are presented in Goldberg [1989], David [1991], Michalewicz [1992], Beasley *et al.* [1993a, 1993b], Bäck [1996], and Mitchell [1996].

Genetic algorithms are characterised by the following elements:

Individuals: representing possible solutions for a particular problem, with features encoded in the chromosome:

Population: a complete set of individuals for the search process;

Chromosomes: basic units of a genetic algorithm, which encode all the information on how each individual is to be constructed;

Genes: subsets of a chromosome. that keep a particular feature of an individual;

Fitness function: the value assigned to each individual, which represents how good the individual is as a solution for the given problem.

Good representation for the chromosomes and, consequently, for each individual is very important for the performance of the genetic algorithm. Traditionally, chromosomes are represented by sequences of genes in terms of binary digits. Nevertheless, other representations are possible; for example, integer or floating-point numbers, letter strings, etc. (Beasley *et al.*, 1993b). The influence of each chromosome representation on the genetic algorithm cannot be quantified or qualified by means of the current theory on genetic algorithms. Some suggestions for chromosome representation may be attained with the *schema theorem.* The schema theorem (Goldberg, 1989 and David, 1991) provides an explanation of how genetic algorithms work. Essentially, the theorem states that particular genes in particular positions of the chromosome have a fundamental influence on the fitness distribution. Those genes and their positions in the chromosome define structures called *schemata* that tend to become dominant in the chromosomes in the population as the genetic algorithm progresses. This suggests that the adopted symbols, or alphabet, to define genes for chromosomes should be defined in such a way to allow the largest number of schemata as possible. Goldberg [1989] postulates that binary representations give the largest number of schemata, but studies with other forms have contradicted this affirmation (Beasley *et al.*, 1993b).

Fitness function representations also have an important effect on genetic algorithms. Ideally, fitness functions should be smooth and regular, so that the distribution of the fitness values of the individuals is approximately uniform. However, such a condition is not possible for many problems of interest (Beasley *et al.*, 1993a), and fitness functions must be defined to avoid excessive local maxima, or a

non-expressive global minimum.

A basic genetic algorithm (Goldberg, 1989), as schematised in Figure 3.1, starts with a randomly initialised population of individuals. Each individual is evaluated by decoding its chromosome and applying the fitness function. With the fitness values for all members of the population, the reproduction phase starts to produce new individuals. New individuals are the result of combining individuals of the original population. Reproduction in a genetic algorithm is commonly carried out by the following operators:

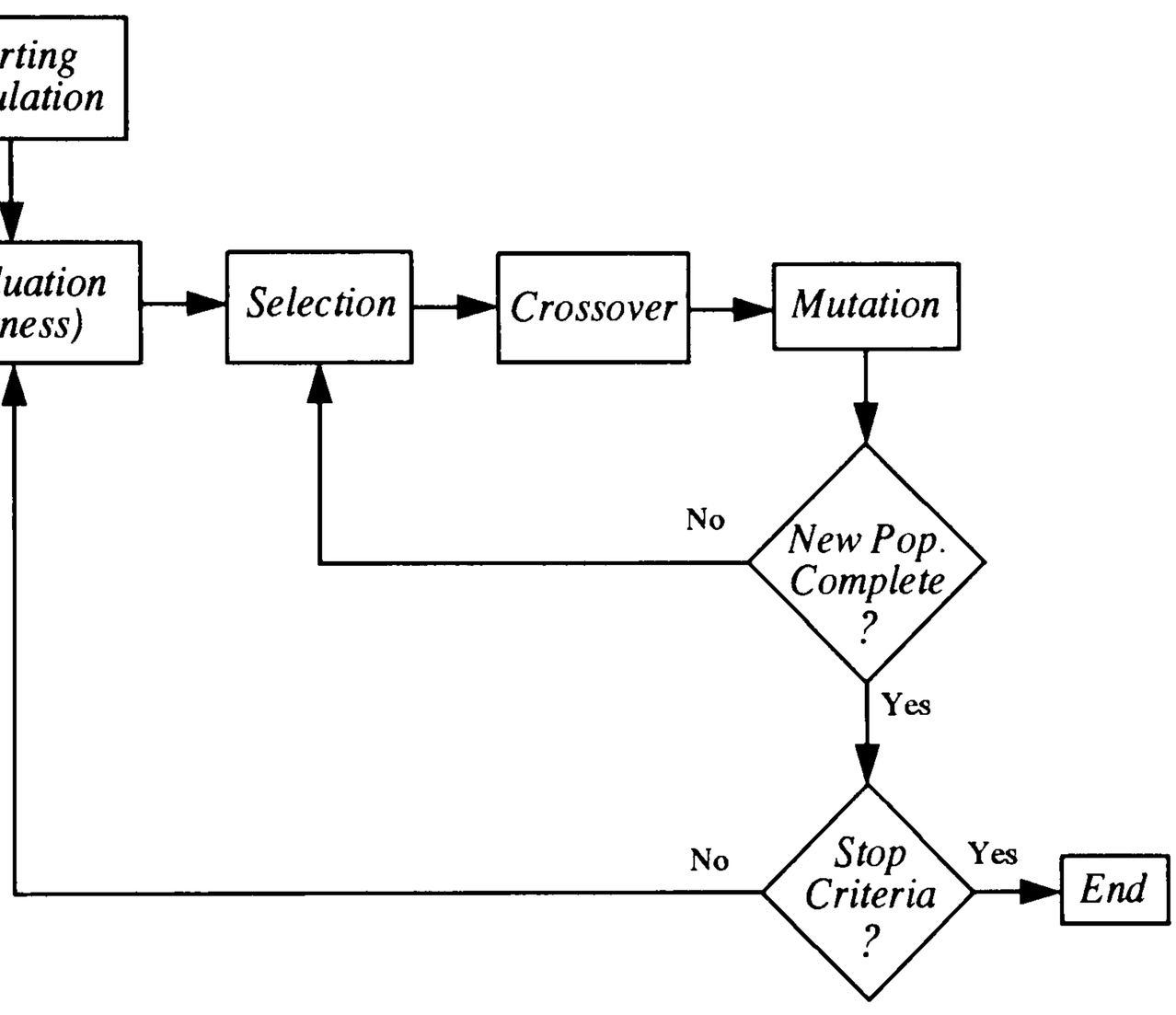


Figure 3.1: Basic genetic algorithm.

Selection: The process in which parents, or pairs of individuals in the population, are chosen for reproduction. Basically, selection is carried out by a variety of random procedures (*roulette wheel selection*, *tournament selection*), and auxiliary mechanisms are adopted to ensure that the fittest individuals have a higher chance

of being selected. The majority of these mechanisms adjust the fitness distribution by scaling or ranking their values.

Crossover: The selection process provides pairs of individuals, namely the parents. The crossover operator proceeds an exchange of genetic material between the parent chromosomes, and as a result, two *new individuals* are created. The two new individuals are likely to be different from each other and their parents, but contain a mixture of genes from their parents. Crossover is not usually applied to all pairs of individuals selected. Typically, only 60% of pairs are subjected to this operator (Goldberg, 1989). The procedure is analogous to what happens in nature. The most basic one-point crossover operation is illustrated in Figure 3.2 and occurs as follows: *(i)* firstly, a randomly assigned gene is set; *(ii)* at this gene position, the chromosomes of each parent are split into two parts; *(iii)* the equivalent parts from each parent are swapped, producing two new individuals. Many other techniques for crossover have been suggested (Beasley *et al.*, 1993a, 1993b). Attempts with multiple-point, uniform, or partially matched crossover have been applied in many studies, but no conclusive arguments exist to establish which is the best method.

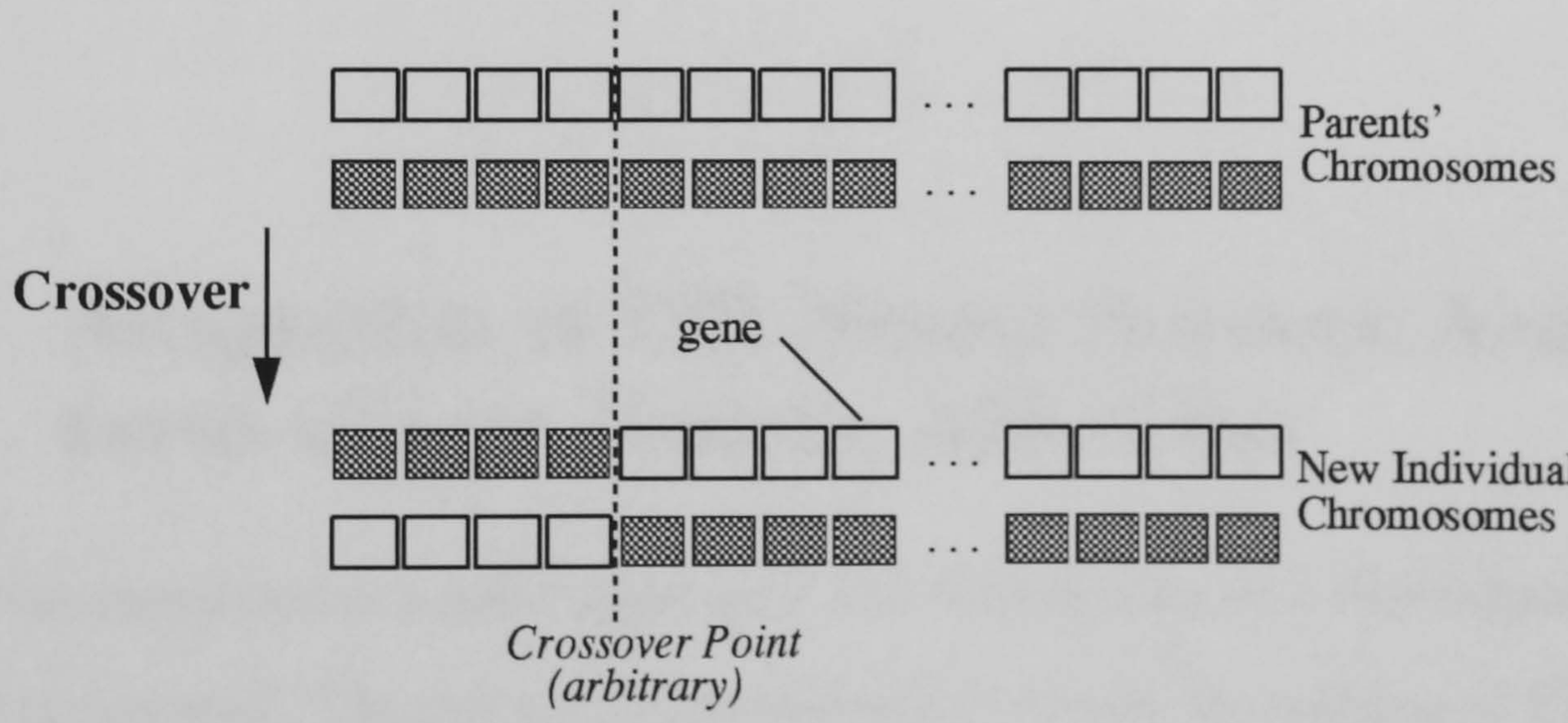


Figure 3.2: The one-point crossover operator.

Mutation: Although mutation has always been associated with a destructive process, it is absolutely necessary for evolution. Genetic algorithms use mutation in many different ways. Its application can be random (with a very low probability value) or systematic (mutation of each new individual). The mutation is commonly made by taking a chromosome of a new individual and, randomly, changing one or more genes. In the context of optimisation, the mutation operator attempts to guarantee that, occasionally, new search spaces can be visited, increasing the chance of avoiding local minima.

Typically, the reproduction process is repeated until the number of new individuals equals the population size. Then, the complete set of new individuals replaces the old population, and the process is stepped a *generation.* Before restarting the algorithm, a stop criteria must be checked. Stop criteria may be based on comparing the highest fitness value with a goal value or by assuming a maximum number of generations. The simplicity and flexibility of genetic algorithms enables several variations and forms. Because theoretical aspects of genetic algorithms are still in their infancy, most of the knowledge about genetic elements and operators relies on empirical works.

3.3 Adaptation of FIR Neural Network Architectures using a Genetic Algorithm

Based on conventional genetic algorithms, the identification of a FIR neural network model is executed. The process simultaneously embraces the training of FIR neural networks weight and bias values, as well as the adaptation of the FIR network architectures.

3.3.1 Encoding FIR Neural Networks

Initially, the genetic elements for the problem must be identified. A FIR neural network is accounted as an *individual* belonging to a *population*, or a set of FIR neural networks. The individuals' *chromosomes* should have *genes* maintaining information such as neurons per layer, time-delays per connection, and weight and bias values.

The fitness function for training and adapting FIR neural network architectures is based on the premise of a *supervised training process*. In supervised training, the neural network is modified depending on the level of error between *actual* and *desired* FIR neural network outputs. Such a strategy is usual for back-propagation training, where the error measurements define a cost function which is used to calculate the gradients.

Here, a cost function, defined as the sum of the squared errors between the desired and the actual network outputs, is adopted. Another important aspect in the fitness function definition is to account for information from several input-output training sets, since non-linear model identification is considered. Therefore, the fitness function f is defined as,

$$f = \left(\sum_{k=1}^{N_c} \sum_{n=1}^{L} (d_k(n) - y_k(n))^2 \right)^{-1} \tag{3.1}$$

where N_c is the number of input-output training sets, L is the total number of time samples per training set, $d_k(n)$ is the desired output of training set k at discrete time n and $y_k(n)$ is the corresponding network output.

The main difficulty in implementing a genetic algorithm to train and adapt FIR neural networks is in encoding a range of possible networks architecture. To determine the chromosome structure, it is first necessary to assume certain features for each FIR neural network (individual). The following assumptions are adopted.

(i) Each network presents a multi-layered architecture of biased neurons without

missing connections between each hidden layer;

(ii) All hidden neurons are non-linear, defined by a sigmoidal activation function, φ, of the activation potential, v; that is,

$$\varphi(v) = \frac{2}{1 + e^{-2v}} - 1 \tag{3.2}$$

(iii) The output neurons have linear activation function, φ, of the activation potential, v; that is,

$$\varphi(v) = v \tag{3.3}$$

(iv) Each connection presents its own number of time-delays, that may differ from connection to connection.

In accounting for all information about the network; for instance, time-delay distributions, neurons, connectivity, and weight-bias values, the chromosome will inevitably be lengthy. Although, there is no restrictions in the genetic algorithm theory about lengthy chromosomes, it is possible that longer chromosomes may reduce the convergence rate of the overall process. To alleviate this effect, the chromosome for the training process is rationalised. Since weight and bias are real values, their inclusion in the chromosome would disrupt the process. A small alphabet should be taken to permit a natural expression of the problem (Goldberg, 1989). Consequently, the chromosomes are comprised of a sequence of the time-delays per connection and the neuron information that determines if the neuron exists or not. It must be borne in mind that the respective values of weight and bias for each delayed connection, and the neuron information, are always related during the genetic operations.

To facilitate the implementation of a conventional crossover operator, the chromosomes are constructed to maintain the same size, whatever the encoded network architecture. Alternatively, the values for genes that identify existent connections or

neurons are encoded. To achieve this, a basic FIR network with bounded architecture, and hence pre-defined chromosome length, is assumed. Then, the chromosome must furnish information about the existence of neurons and their connections, as well as the values of time-delays per connection.

The sequence of genes in the chromosome is based on the number of neurons of the bounded network. The genes representing the neuron existence values are enumerated from the first neuron in the nearest hidden layer to the input layer, to the last neuron of the output layer. For each neuron of the bounded network, the respective gene is a flag to assign whether the neuron exists or not. The genes for time-delays are arranged in terms of the connections between neurons of the network - they are integer numbers that correspond to the respective time-delay values. However, in the case of a non-existent connection, a special flag is assigned to the time-delay gene in question. Then, the final sequence for a chromosome is the one in which the flag gene per neuron is preceded by the time-delay genes of all connections to the respective neuron.

Figure 3.3 depicts a generic representation for the chromosome to encode FIR neural networks, where N_i^f is the flag indicating whether the neuron i exists or not, $\tau_{i,j}$ is the time-delay value in the connection between neuron j (in the previous layer) to neuron i, and m is the number of neurons of the previous layer (consequently, if the previous layer is the input layer, m is equal to the number of inputs).

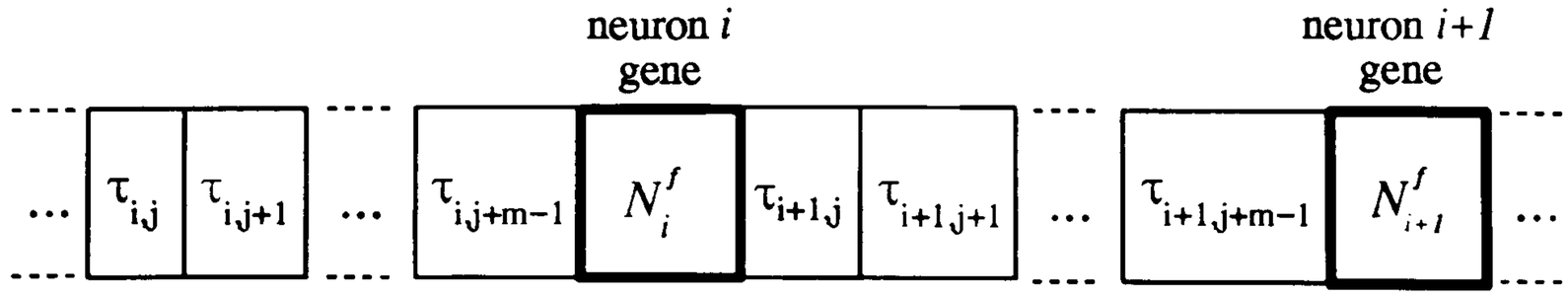


Figure 3.3: Generic representation of the chromosome.

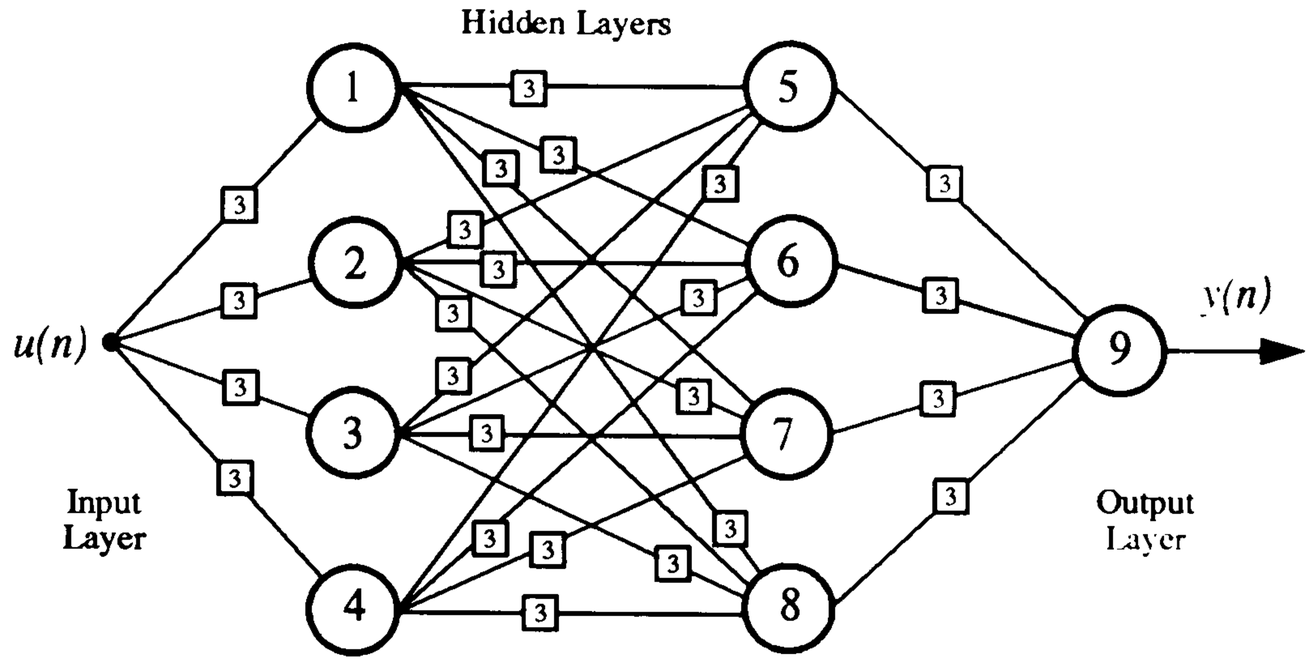
An example of how a chromosome encodes FIR neural network architectures is presented in Figures 3.4 (a) to (c). Figure 3.4 (a) shows an arbitrary maximum possible architecture for a single input single output FIR neural network. This example considers each connection possessing a maximum of three time-delays ($\tau_{max} = 3$), with the neurons disposed in two hidden layers. The neurons are represented by the enumerated circles, while the squares represent the time-delays in the respective connection. Figure 3.4 (b) shows one of the possible FIR neural network architectures that can be extracted from the bounded network architecture (*cf.* Figure 3.4 (a)), while its respective chromosome is depicted in Figure 3.4 (c). In this example, the neurons 3, 5 and 7 do not exist; accordingly the flag F (false) is assigned in the chromosome. In contrast, existent neurons receive the flag T (true). The non-existent connections receive a special flag, represented here by the symbol *. The values for the time-delays (depicted by connection) are arbitrary, and for the connections without values, the time-delay is considered null.

3.3.2 Genetic Operators

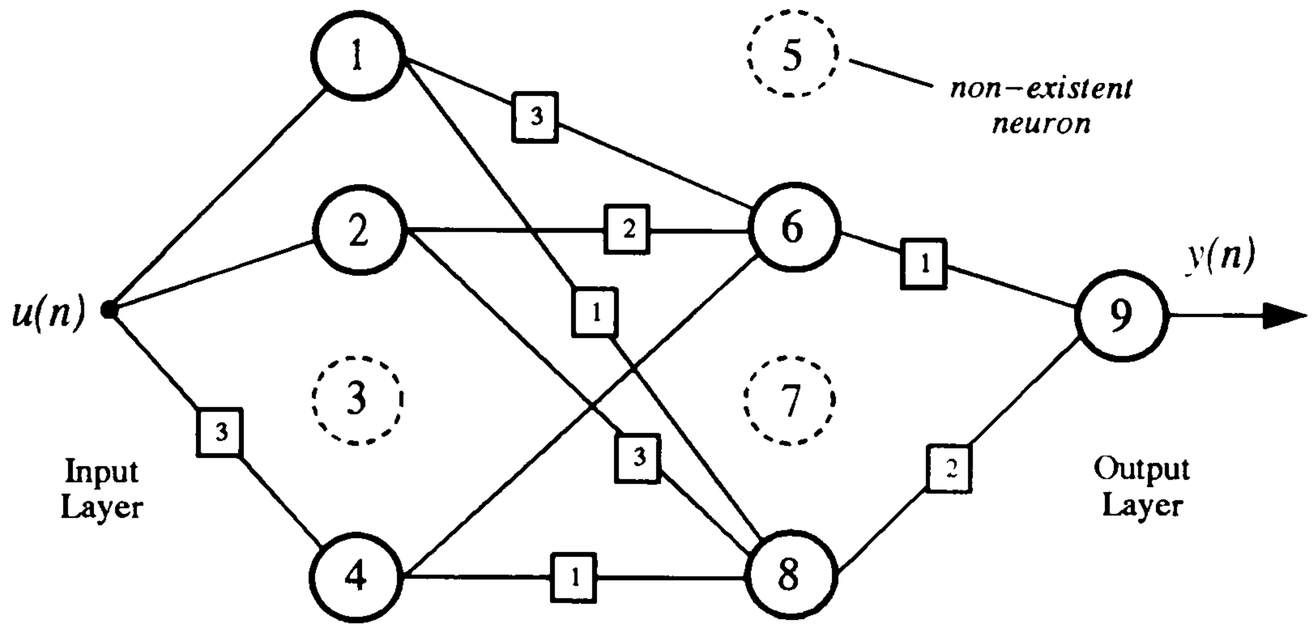
Essentially, the genetic operators used for the training and adaptation of FIR neural network architectures are based on the conventional operators described in Section 3.2. A description of each of the genetic operators follows.

Selection

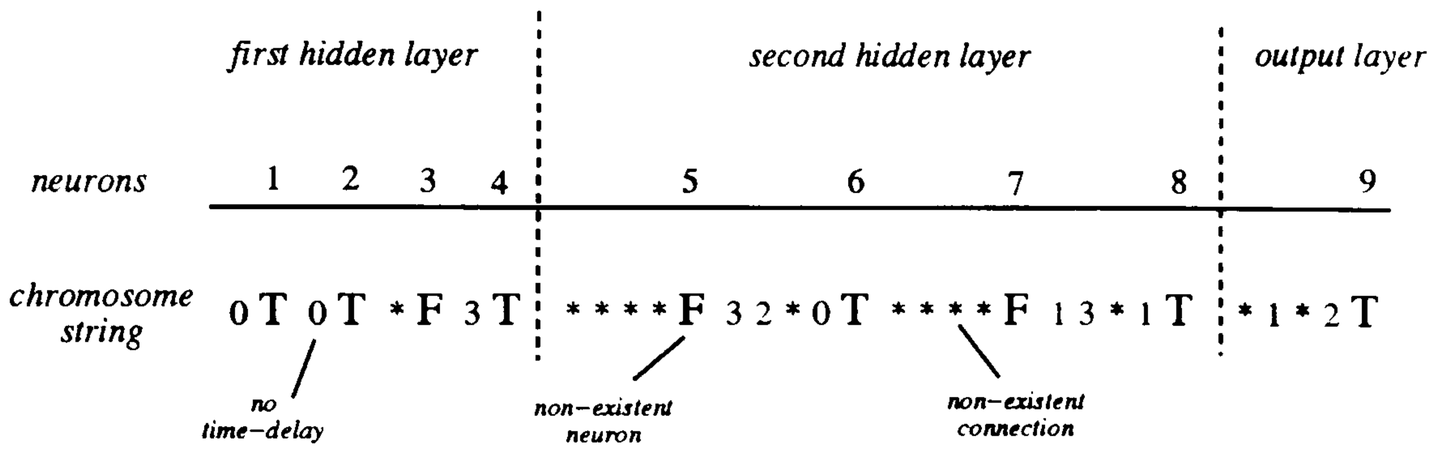
The selection operator is based on the *roulette wheel selection* described by Goldberg [1989], combined with a scaled fitness distribution. The roulette wheel selection proceeds a random choice for parents starting by summing all fitness values in the population and multiplying the result by a uniformly distributed random number between 0 and 1. This produces a *reference value*. Then, the chosen individual (par-



(a) Maximum FIR network architecture



(b) Extracted FIR network architecture



(c) Chromosome representation

Figure 3.4: Example of encoded FIR neural network architecture.

ent) is the one whose fitness, added to the fitness values of the preceding individuals, is greater or equal than the *reference value*. The second individual (second parent) for reproduction is chosen by repeating the process for a different random number, and consequently, a different *reference value*.

The application of pure roulette wheel selection, however, may not be so efficient, because the process offers the same chance of selection for all the members of the population. In addition, for the first generations the process is very likely to encounter individuals with much different fitness values, from a very fit to an extremely poor one. Possibly, individuals with poor fitness value may present good genes in their chromosomes, but it is convenient to avoid them and to establish a way of accelerating the convergence rate without weakening the quality of the final process outcome. A common practice is to concentrate the selection on the fittest individuals of the population, by scaling the fitness distribution.

Here, a linear *scaling* process is applied (Goldberg, 1989). The calculation of the new scale is based on the average fitness value in the population. Defining f as the actual fitness of an arbitrary individual of the population (*cf.* Equation 3.1), its value in the scaled fitness is given by: $f^s = af + b$, where a and b are coefficients which depend on the *minimum*, *average* and *maximum* actual fitness values in the population and s denotes scaled fitness values. The average values of both actual and scaled fitness are assumed to be the same to guarantee that each average population member contributes to the next generation. The maximum scaled fitness must be proportional to the actual average fitness, given by, $f^s_{max} = c\, f_{avg}$, where c determines the number of times the best individual is expected to be selected. A typical value for the coefficient c is between 1.2 and 2 (Goldberg, 1989).

The scaling process, however, may present a situation in which few individuals with very poor fitness values are far below the population average. So, in applying the scaling, those lower fitness individuals may become negative, thereby spoiling

the selection process. This situation is likely to happen after the population fitness values start to stabilise. Goldberg [1989] shows how these problems can be tackled. A test is carried out during the scaling. If the minimum scaled fitness is predicted to be negative by normal scaling, then a new condition is set. It is assumed that the average scaled fitness continues to be equal to the actual one, but the minimum scaled fitness will be set zero. The final forms of the coefficients a and b for scaling the population's fitness values are given in Table 3.1.

$f_{min} \leq \frac{cf_{avg}-f_{max}}{c-1}$	$f_{min} > \frac{cf_{avg}-f_{max}}{c-1}$
$a = \frac{(c-1)f_{avg}}{f_{max}-f_{avg}}$	$a = \frac{f_{avg}}{f_{avg}-f_{min}}$
$b = \frac{f_{avg}(f_{max}-cf_{avg})}{f_{max}-f_{avg}}$	$b = \frac{-f_{min}f_{avg}}{f_{avg}-f_{min}}$

Table 3.1: Scaling coefficients a and b for the selection operator.

Crossover

The only special feature of crossover, which is used in the genetic algorithm applied to FIR neural network training, is the adoption of multiple-point crossover. Since chromosomes encoding FIR neural network architectures can be lengthy, the adoption of many points for this operation is attempted in order to improve the convergence performance.

Nevertheless, a drawback associated with multiple-point crossover is the loss of many good genes when high numbers of points are taken. A reasonable number of points to avoid this sort of disruption can only be discovered by experimentation.

Mutation

For each new individual created, the mutation operator randomly changes each gene of the chromosome, based on a certain low probability value for mutation. As the chromosomes present particular symbols (alphabet), which determine features of the FIR neural network architecture, a coherent mutation operator has to be adopted. Here, mutation is guided by the probability value P_t, for the mutation of time-delays, or P_n for the mutation of a neuron value (existent or non-existent), as the operator is presented with each of the chromosome genes. The mutated genes must be in the specified range of values, and any change in time-delays or neuron existence is followed by a respective change in the weight and bias values. The mutation of an existent neuron leads to the complete disappearance of the connections reaching and leaving the neuron. In contrast, where non-existent neurons are mutated, all connections to and from this new neuron are created.

3.4 A Supervised Training Process for Network Identification

The basic genetic algorithm (*cf.* Figure 3.1), with elements and operators as described in Section 3.2, is used for the supervised training and architecture adaptation of FIR neural networks.

The process commences with an initial population of individuals of size P. Each individual is created with a randomly generated architecture, where the time-delay values are taken between zero (connection without delay) and the maximum prescribed value τ_{max}, and the neuron flags receive a value which determines their existence (50% of existence). Depending on the time-delays and neuron flags, the

respective uniformly distributed random weight and bias values (between -1.0 and 1.0) are assigned.

Then, the entire population is evaluated by a feedforward pass of each individual (FIR network), for each one of the pre-defined training sets, using Equation 3.1. The fitness distribution of the population is used to promote the parents' selection for reproduction. As described in the previous section, selected parents produce new individuals through crossover and low probability mutation operators.

Some care must be taken after the creation of new individuals, or after their mutation. Encoded FIR neural networks of different architectures may present problems during genetic operations. The gaps left inside the chromosomes by non-existent neurons or connections may lead to an inconsistent new individual after the crossover operation. The procedure must be monitored to identify potential anomalies using checking routines applied just after the creation or mutation of new individuals. An attempt is made to correct any distortion in the architecture of new individuals, thereby, enabling chromosome strings to be re-arranged in the best way possible. If this is not feasible, then the new individual is discarded.

The next step after the main genetic operations is to update weight and bias values of the new individuals pairs. Because of the nature of these values (real numbers) and the structure adopted by chromosomes, their manipulation is more conveniently carried out through another operation. This operation consists of perturbing the weight, $\mathbf{w}$, and bias, $\boldsymbol{\theta}$, values by a scaled, zero mean, unit variance normally distributed random value, $\mathbf{N}(0,1)$; that is,

$$\begin{aligned} \mathbf{w}_{new} &= \mathbf{w}_{old} + \beta\mathbf{N}(0,1) \\ \boldsymbol{\theta}_{new} &= \boldsymbol{\theta}_{old} + \beta\mathbf{N}(0,1) \end{aligned} \tag{3.4}$$

where β is a small proportionality constant.

This process is repeated several times for the same individual, and for each the

new values of weight and bias are only accepted if they lead to a fitter individual. Then, the modified individual is returned to the population. In essence, this process randomly investigates vicinities of the FIR network weight and bias values for perturbations that lead to a reduction of the cost function (or to an increase of the fitness value). The intensity of the perturbation is controlled by the constant β, which also influences the speed of convergence, since large perturbations also mean larger differences in the weight space which in term increase the chance of disruptive deviations from optimum neighbourhoods. A suitable value for β can only be adopted after some experimentation with the algorithm and the respective input-output mapping. In addition, this random search process may be associated with a crude implementation of the *simulated annealing algorithm* (Kirkpatrick *et al.*, 1983; Rutenbar, 1989 and Otten & van Ginneken, 1989), where a cooling schedule is not accounted for.

The algorithm continues to create new individuals and to update their weight and bias values until the number of new individuals equals the original population size P (one generation is stepped). Following each generation, the new and old individuals, totalling $2P$ individuals, are compared in terms of their fitness values and only the best P individuals are retained for the next generation. Therefore, as there is no complete replacement of the old population, it is possible that the same best individual occurs for many generations. This methodology has been adopted to avoid the loss of good *genetic material* from one generation to another, and to reduce potential disruptions in the overall process.

To mitigate against the possibility of convergence to sub-optimal solutions, an additional operation is used. Occasional stagnation of the process has been observed during the test phase of the algorithm. Generally, this effect happens when the population has reached a stable condition in a sub-optimal solution, and no significant improvement occurs to overcome this situation. The need for new information in

the population to modify the search space is clear.

When stagnation occurs, the algorithm forces the whole population to mutate in response to another mutation operator, referred to as *forced mutation*. This operator is invoked only after a pre-defined number of generations present the same fitness value for the best individual, and this operator works differently from the mutation operator applied after the creation of new individuals in the genetic algorithm. For each member of the population, the operator randomly selects a gene to be modified. Then, the gene is modified and the individual is tested to check if its new fitness value is greater than the old one. If this is the case, the individual is accepted, otherwise it is only accepted with a fixed probability value of 0.0001. This probability value has been adopted after some numerical experiences with the training algorithm. Again, this operator works analogously to the conventional simulated annealing algorithm, but here the cooling schedule presents a fixed temperature value and no stabilisation inner loop. The routine is repeated a number of times and the final mutated individual returns to the population, restarting the whole procedure for the next member of the population.

The final stage of the training and adaptation algorithm is a check for terminating the process. Two criteria are used to terminate the algorithm; namely *(i)* the number of generations exceeds a pre-defined limit, and *(ii)* the fitness value of the best individual exceeds a pre-defined goal value. If any of the aforementioned criteria are not met, the algorithm restarts to evaluate the population, followed by genetic operations and so on. Figure 3.5 presents a schematic representation of the algorithm to illustrate the phases of the training process.

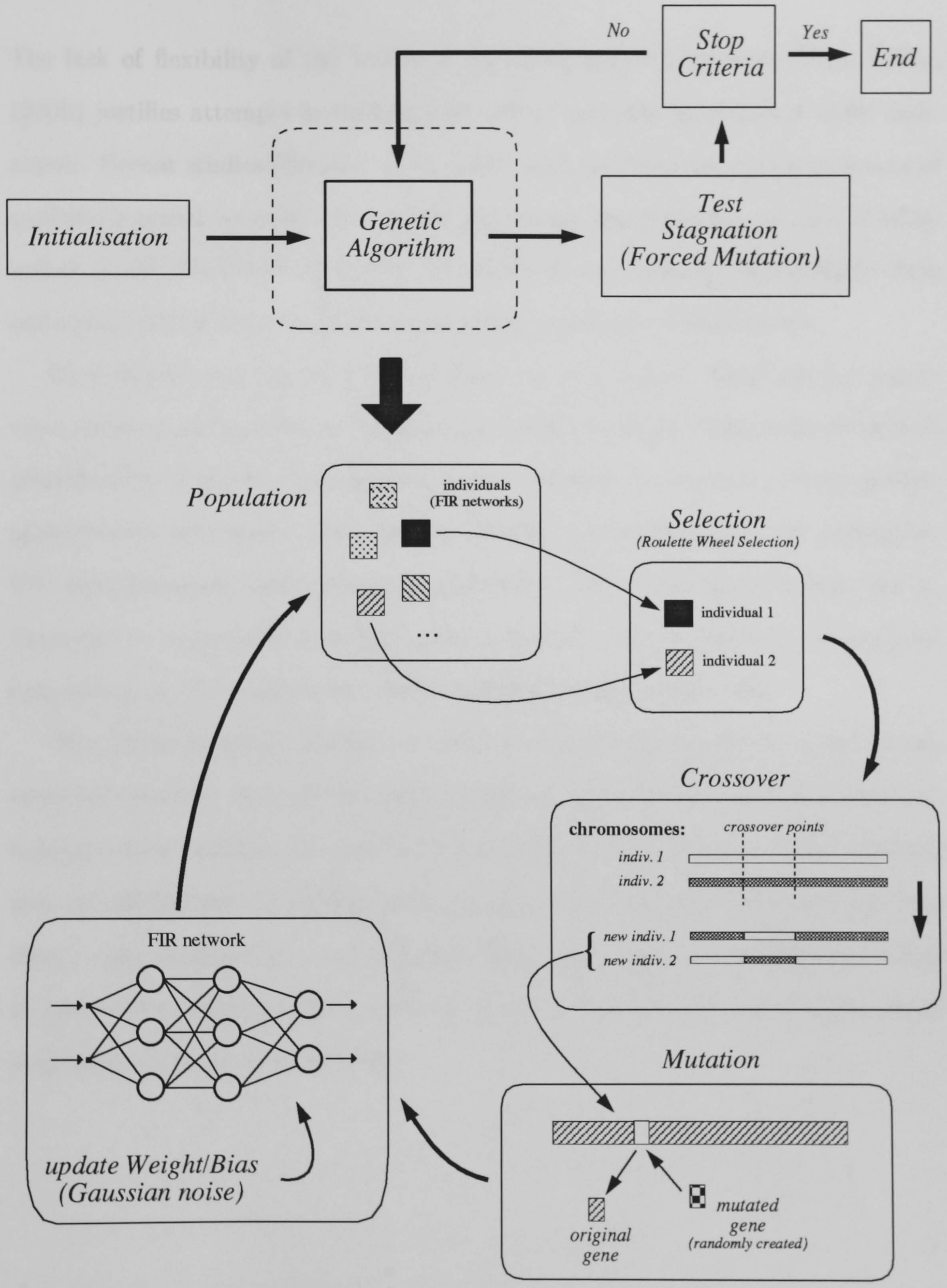


Figure 3.5: Schematic representation of the training algorithm.

3.5 Summary

The lack of flexibility of the temporal back-propagation algorithm (Wan, 1990a, 1990b) justifies attempts in training FIR neural networks by means of other techniques. Recent studies (Schaffer *et al.*, 1992) have provided arguments in favour of combining neural networks and genetic algorithms, especially for the case of adaptation of neural network architectures. Therefore, the use of genetic algorithm to train and optimise FIR neural network architectures seems to be appropriate.

This chapter has described the application of a genetic algorithm for supervised training and architecture adaptation of FIR neural networks in the context of identification of non-linear dynamic systems. Initially, an overview of basic genetic algorithms is presented. Then, genetic elements and operators for the adaptation FIR neural network architectures are described. The chromosomes encode the information to reconstruct each FIR neural network in the population, allowing the connections in the networks to present different values of time-delay.

The genetic operators are the conventional ones (Goldberg, 1989), except for the crossover operator, that permits multiple points. The selection operator is based on roulette wheel selection over a scaled fitness distribution, and the mutation operator uses two probability values depending on the gene to be mutated (neuron or time-delay). The algorithm also uses simplifications of the simulated annealing algorithm to update the weight and bias values of the FIR neural network, as well as to avoid stagnation of the convergence rate.

Chapter 4

Modelling Non-Linear Unsteady Aerodynamic Response by Multi-Layer Functional Approximation

4.1 Introduction

The objective of this chapter is to address the issues and achievements obtained with the application of multi-layer functionals in the identification of non-linear unsteady aerodynamic response models. Multi-layer functionals, characterising the non-linear relationship between the (generalised) motion history and the (generalised) unsteady aerodynamic response, are implemented in terms of a discrete-time version of the temporal neural network; the FIR neural network. Identification of the FIR neural network is carried out using a process based on genetic search, in which both the network training and optimisation of the network architecture is established.

Ideally, practical implementations of multi-layer functional representations of non-linear unsteady aerodynamic response for aeroelastic analysis and control design should be capable of representing a large class of flow regimes under normal operating

conditions. However, due to the physical characteristics of unsteady aerodynamic response; for example, flow separation and compressibility effects, multi-layer functionals in the form of FIR neural network are specific to particular ranges of flow regimes and aerodynamic geometries. In addition, limitations due to the inherent characteristics of multi-layer functionals, restrict applications to motion histories compatible with the corresponding attached to mildly separated flow non-linearities of the unsteady aerodynamic response.

Indeed, for such flow conditions, the unsteady aerodynamic reponse can be described mathematically as a continuous functional of the motion variables compatible with the properties of multi-layer functionals. Therefore, the following multi-layer functional representation can be associated with the unsteady aerodynamic response model,

$$\mathbf{F}(t) \approx \mathcal{MF}[\mathbf{u}_t, M] \tag{4.1}$$

where $\mathbf{F}(t)$ is the unsteady aerodynamic force vector, $\mathcal{MF}$ is the multi-layer functional operator, $\mathbf{u}_t$ is the input motion history vector and M is a flow parameter; for example, the Mach number.

The unsteady aerodynamic test cases considered here are associated with two types of flow regimes: *(i)* mildly separated flowfields, and *(ii)* compressible flows exhibiting dynamic motion of shock waves. In the first case, weakly non-linear behaviour of the unsteady aerodynamic response is observed, while in the second, a stronger non-linear behaviour occurs. For the mildly separated flow regime case, the aerodynamic data necessary for identification is obtained by using a computational implementation of the semi-empirical Beddoes model by Niven & Galbraith [1991]. The transonic aerodynamic database is created using a CFD code (Dubuc *et al.*, 1997) based on the numerical solution of the Euler equations. In both cases, a two-

dimensional NACA 0012 airfoil is used, and the range of motion histories is selected to cover a large class of non-linear unsteady aerodynamic behaviour compatible with the limitations of the database generators.

The training process demands that a broad range of motion induced unsteady aerodynamic responses be used during the non-linear identification of the FIR neural network. This requirement is associated with the non-linear nature of the unsteady aerodynamic response of interest, and the need for a variety of characteristic motion histories within the chosen range for the network identification process. Other flow parameters can also be considered to vary from one motion history to another. For instance, a static range of Mach numbers variations per motion history can also be accommodated in the aerodynamic database.

This chapter presents a description of the computations used to build up the aerodynamic database required for network training. A brief description of the Beddoes model and the Euler CFD code is made, and the respective numerical experimentations developed to establish the non-linear range of the unsteady aerodynamic responses. The results of model identification for both cases and a discussion are also presented.

4.2 Unsteady Aerodynamic Data Generation

The data base generation used in the identification of non-linear unsteady aerodynamic response functionals is presented. Only flow regimes that can be represented by continuous functionals are considered. Discontinuities in the functional representations are related to the effects of separation. As shown by Tobak & Chapman [1985], functionals for this type of flow regime have to account for the replacement

of an unstable equilibrium flow by a new stable equilibrium flow at a critical condition; in other words, account for the effects of aerodynamic bifurcation. Subjected to these constraints, weakly non-linear unsteady aerodynamic behaviour associated with mildly separated flows, and non-linear inviscid unsteady aerodynamic behaviour in the transonic regime, are considered.

The approaches employed here to generate the unsteady aerodynamic database are: *(i)* the Beddoes model (Niven & Galbraith, 1991) for weakly non-linear unsteady aerodynamic response of mildly separated flowfields; and *(ii)* an Euler CFD code (Dubuc *et al.*, 1997) for unsteady aerodynamic response in the transonic regime.

4.2.1 A Semi-Empirical Model for Mildly Separated Flow

The Beddoes model is a semi-empirical model originally proposed by Beddoes [1976, 1982a, 1982b] and Leishman & Beddoes [1986]. This approach uses concepts based on physical observation of overall unsteady aerodynamic behaviour of bi-dimensional airfoils to determine a mathematical and logical formulation of the relationships between the motion history and the aerodynamic response. The model produces aerodynamic reactions by using superposition of a series of linear indicial responses. Extension to non-linear unsteady behaviour is accounted for by specific theoretical or experimental correlation. Some of the most important evaluation aspects of the various unsteady features executed in the Beddoes model (Beddoes, 1976, 1982a, 1982b and Leishman & Beddoes, 1986), are, briefly: *(i)* non-linear lift characteristics of trailing edge separation are calculated using a Kirchhoff model (*cf.* Appendix A); *(ii)* centre of pressure variation with separation is empirically evaluated, based on the airfoil static behaviour; *(iii)* delays are added to pressure variations and unsteady boundary layer response; *(iv)* the dynamic trailing edge separation point is computed and used as a primary degree-of-freedom in the model. The complete model is

capable of monitoring local flow states during unsteady motion to combine the temporal and non-linear effects to produce the final aerodynamic response.

The weakly non-linear behaviour of the unsteady aerodynamic force response of a two-dimensional NACA 0012 airfoil in mildly separated flows for a range of Mach numbers is considered for the identification process. Under these conditions, the Beddoes model uses the Kirchhoff flow model, as presented in Appendix A.

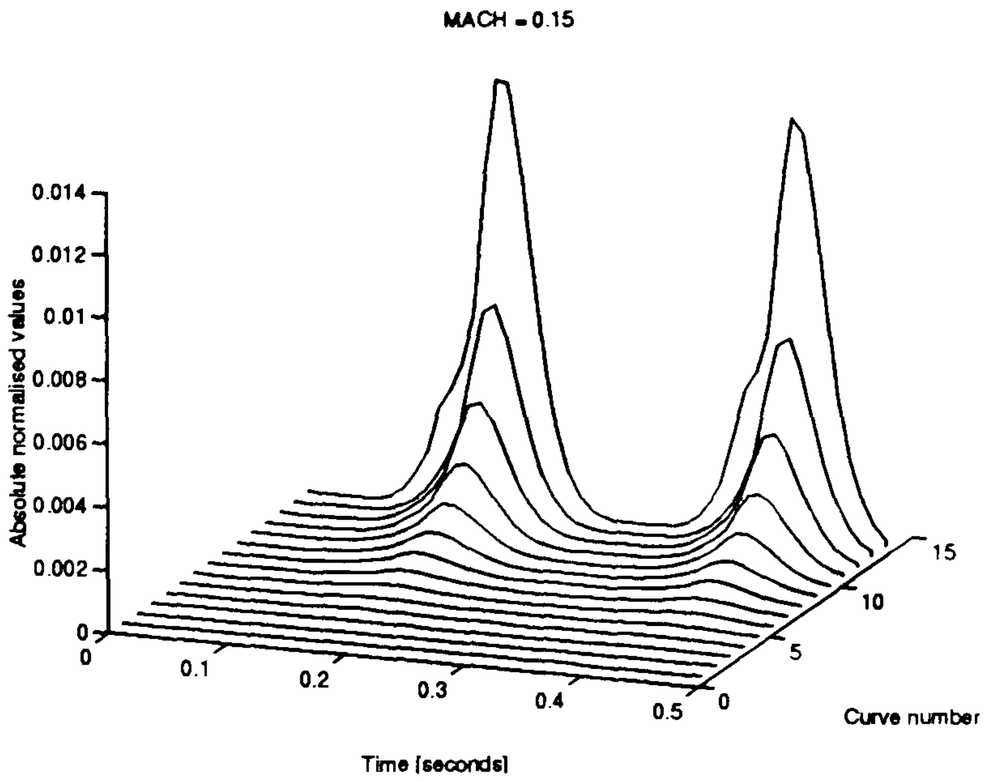
Niven & Galbraith [1991] have coded the Beddoes model into a computational program, which is used to generate the aerodynamic database. Numerical experimentation is used to indicate the range of incidence angle (angle of attack) motion histories for weakly non-linear behaviour of unsteady aerodynamic normal force response. A freestream sonic velocity of 340.5 $m.s^{-1}$ and an airfoil chord length of 0.55 m are adopted. The non-linear behaviour is studied from the numerical experiments by observing the differences between consecutive aerodynamic responses for the same type of incidence motion history at an increasing maximum value. For Mach numbers; 0.15, 0.30 and 0.45, sinusoidal incidence histories of the airfoil at a fixed frequency of 2 Hz and zero mean angle, are considered. From an initial amplitude of 1°, up to the maximum amplitude value of 15°, the incidence histories stepping 1° in the amplitude value are used to obtain the respective normal force coefficient response histories, $C_N(t)$.

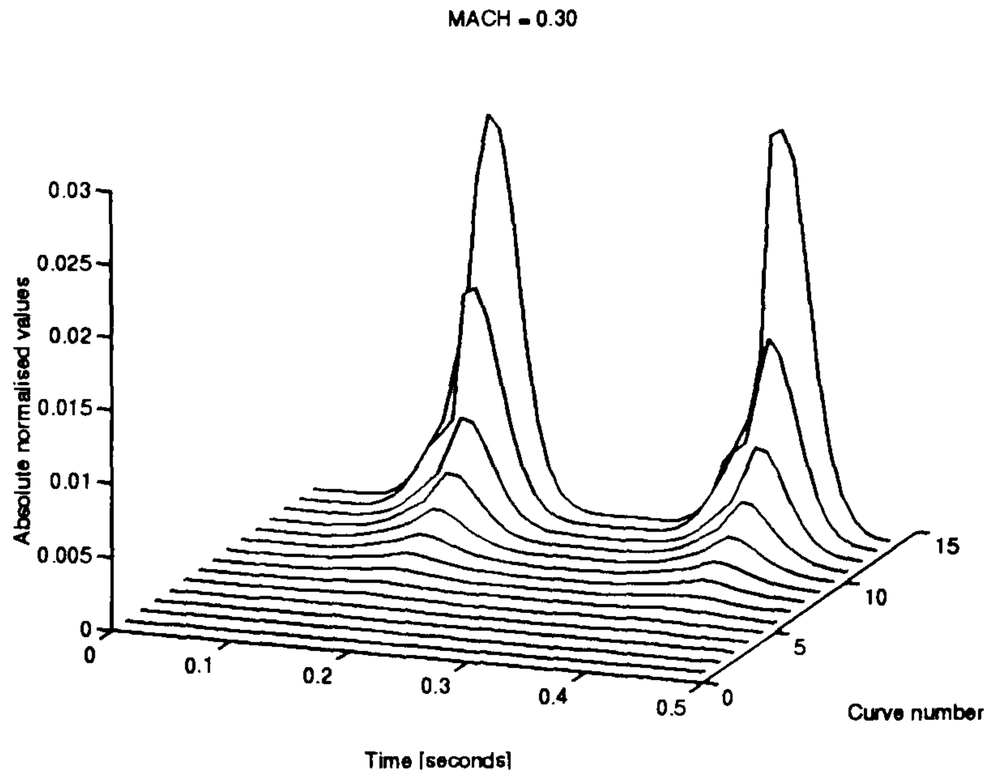
Then, the difference between each $C_N(t)$ response history is calculated as,

$$\frac{C_{N_{A+1}}(t) - C_{N_1}(t)}{A} \tag{4.2}$$

where $C_{N_{A+1}}(t)$ is the response history due to the motion of amplitude $A + 1$ and the $C_{N_1}(t)$ is response history due to the 1° amplitude motion, for $A = 1°, \ldots, 15°$.

This procedure generates a family of $C_N(t)$ difference curves that are normalised with respect to the first $C_N(t)$ difference curve. The result is a family of normalised $C_N(t)$ difference curves as shown in Figure 4.1 for each respective Mach number.





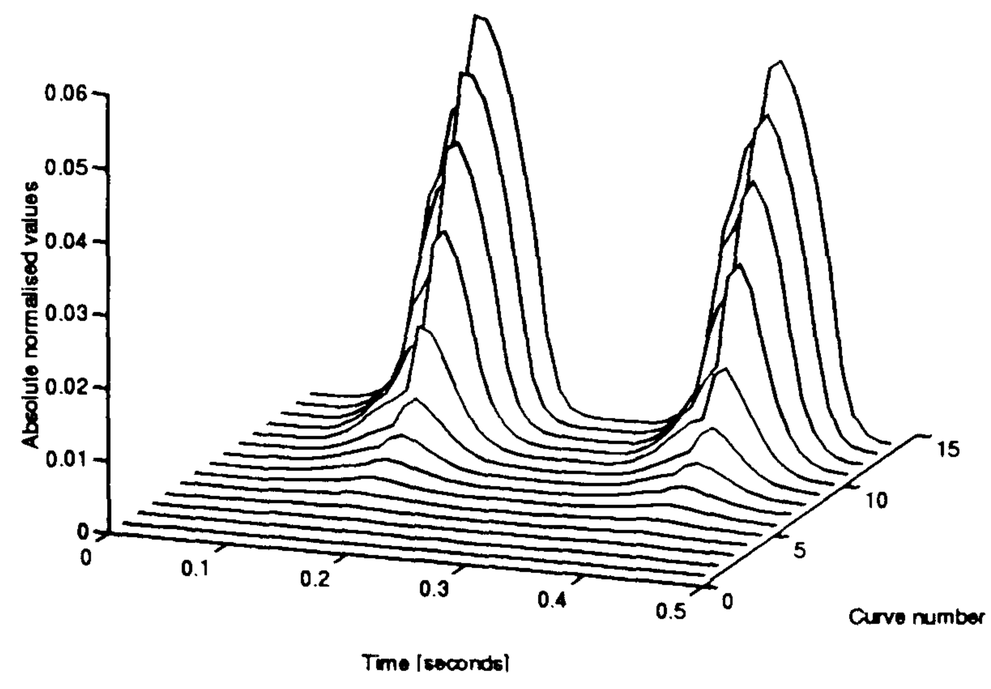


Figure 4.1: Weakly non-linear behaviour of $C_N(t)$.

The non-linear behaviour of the normal force coefficient responses can be extracted from the normalised $C_N(t)$ difference curves presenting non-zero absolute normalised $C_N(t)$ values. Inspection of Figure 4.1 reveals the range of angles of attack (incidence motion amplitudes) where the weakly non-linear behaviour induces non-zero $C_N(t)$ difference curves, and Table 4.1 summarises these values.

Mach numbers	Range of absolute incidence angles
0.15	8° *to* 13°
0.30	7° *to* 12°
0.45	6° *to* 10°

Table 4.1: Range of angles of attack that induces weakly non-linear unsteady aerodynamic behaviour.

4.2.2 An Euler Model for Transonic Flow

The CFD code developed by Dubuc *et al.* [1997] is used for the unsteady transonic aerodynamic database generation. The code solves the non-linear unsteady Euler equations for inviscid fluid flow by means of implicit methods. An implicit dual-time method is employed to solve the unsteady fluid mechanics equations, while an implicit unfactored method solves the associated steady state problem in pseudo-time. The solution also relies on a multi-block structured grid approach based on a higher-order finite volume discretisation. For the aerodynamic data presented here, an undeformed grid rotational motion is considered. Figure 4.2 depicts a detail of the mesh used by the CFD code.

Unsteady transonic aerodynamic responses, influenced by non-linearities caused

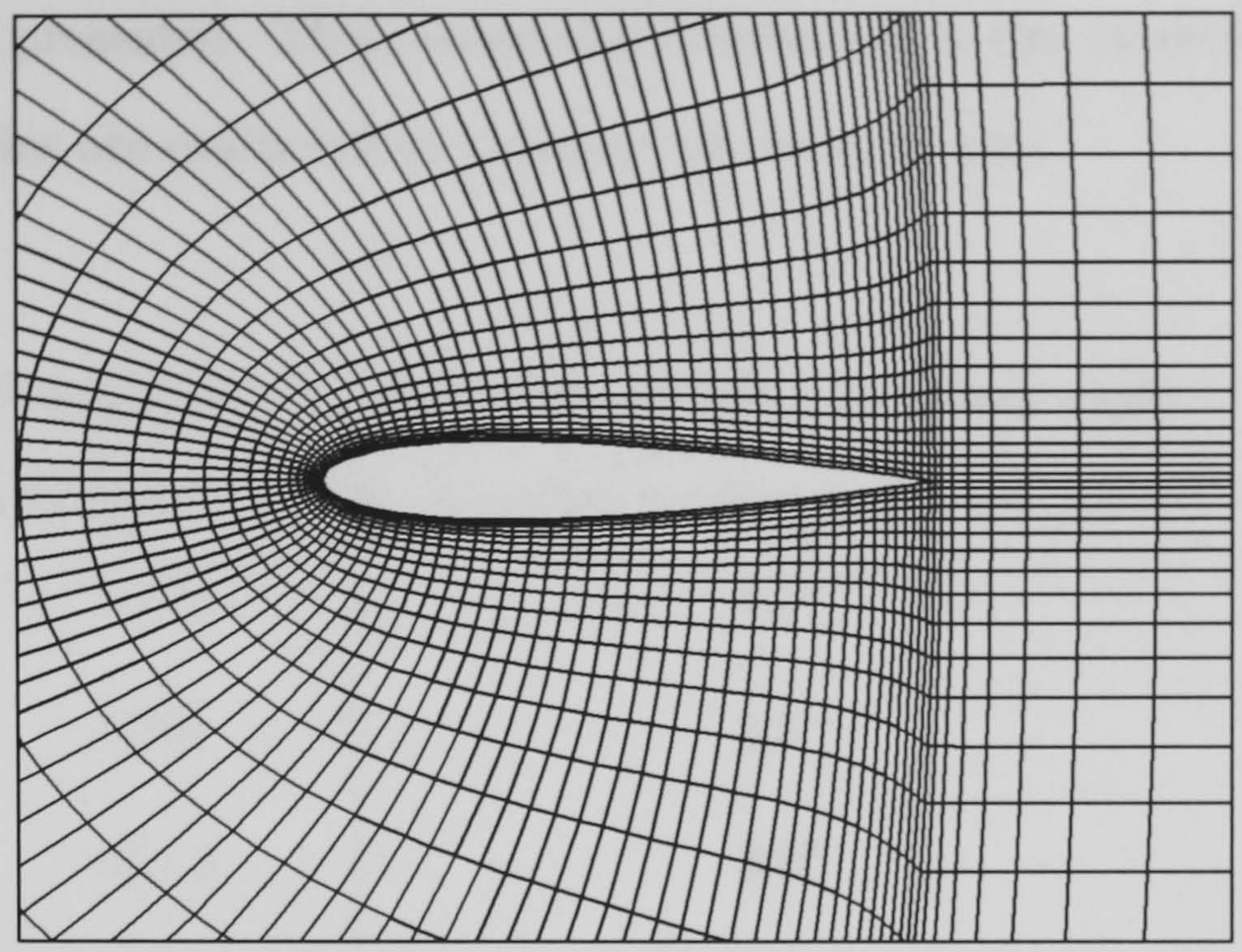


Figure 4.2: Detail of finite volume mesh used by CFD code.

by the appearance and dynamic excursion of shock waves during the incidence motion history, are considered in the database for the training process. To establish the incidence motion and Mach number range that covers the non-linear behaviour of unsteady transonic responses, numerical experiments with a two-dimensional NACA 0012 airfoil, freestream sonic velocity of 340.5 $m.s^{-1}$ and airfoil chord length of 0.55 m, have been carried out.

The experiments are based on observations of variations of the pressure coefficient distribution around the airfoil due to shock waves during unsteady incidence motions. For several combined Mach numbers and motion histories adopted for the training, the maximum (absolute) angle during the motion history exhibiting compressibility non-linearities in the pressure distribution, is assumed as the unsteady motion boundary of the training process at the respective Mach number. Table 4.2 shows the relation between Mach number and the maximum incidence angle for the non-linear unsteady transonic database generation. Moreover, in Appendix B, the pressure distribution variations for the cases used in the training process and further

verifications are presented. These cases also illustrate how the maximum incidence motion boundaries are extracted to compose the training sets.

Mach numbers	Maximum absolute incidence angle
0.625	4.5°
0.65	4.0°
0.675	3.0°
0.725	1.5°

Table 4.2: Incidence angles for non-linear behaviour of unsteady aerodynamic response in the transonic regime.

4.3 Approximation of the Unsteady Aerodynamic Response in Mildly Separated Flow

The identification of a multi-layer functional in the form of a FIR neural network model for the weakly non-linear behaviour of unsteady aerodynamic response in mildly separated flows is presented. The process of network training and architecture adaptation, as described in Chapter 3, is used for the model identification. The algorithm is carried out for a two input one output unsteady aerodynamic problem. The functional relationship between normal force coefficient, $C_N(t)$, and angle of attack history, α_t, is represented by a multi-layer functional, $\mathcal{MF}$; that is,

$$C_N(t) = \mathcal{MF}[\alpha_t, M] \tag{4.3}$$

The data used to train and adapt the FIR neural network is obtained by the semi-empirical Beddoes model presented in the Section 4.2.1. The model implicitly accounts for the Mach number dependence of the unsteady aerodynamic force response, as well as the notional non-linear functional relation between the aerodynamic force response and the incidence motion history, with basic problem parameters as illustrated in Figure 4.3, freestream sonic velocity of 340.5 $m.s^{-1}$, and airfoil chord length of 0.55 m.

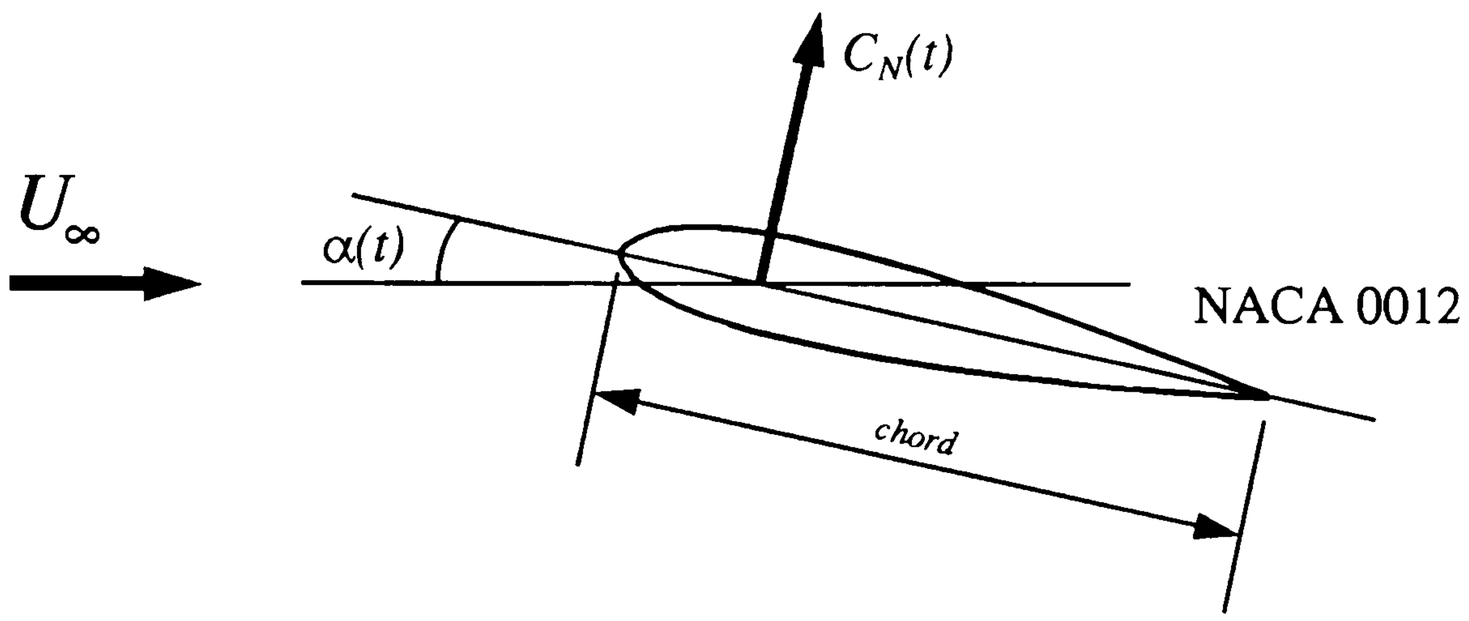


Figure 4.3: Parameters for the identification of the unsteady aerodynamic response multi-layer functional in mildly separated flow.

According to the features of weakly non-linear behaviour of the unsteady aerodynamic response in mildly separated flow, obtained by numerical experiments in Section 4.2.1, characteristic motion histories for the FIR network identification can be defined. The characteristic motion histories comprise sinusoidal, ramp-up, and ramp-down motions of the angle of attack, with maximum absolute amplitudes equal to the maximum absolute angle of attack that determines the limit for weakly non-linear behaviour in mildly separated flow of the unsteady normal force response (*cf.* Table 4.3).

Characteristic motion	M	Range
ramp-up	0.15	$\alpha_{min} = -5°$: $\alpha_{max} = 13°$
	0.30	$\alpha_{min} = -5°$: $\alpha_{max} = 12°$
	0.45	$\alpha_{min} = -5°$: $\alpha_{max} = 10°$
sinusoidal	0.15	$\alpha_{mean} = 0°$: amplitude= 13°
(all cases: frequency is 2 Hz)	0.30	$\alpha_{mean} = 0°$; amplitude= 12°
	0.45	$\alpha_{mean} = 0°$: amplitude= 10°
ramp-down	0.15	$\alpha_{max} = 13°$; $\alpha_{min} = 0°$
	0.30	$\alpha_{max} = 12°$; $\alpha_{min} = 0°$
	0.45	$\alpha_{max} = 10°$: $\alpha_{min} = 0°$

Table 4.3: Training set motions for the identification of the unsteady aerodynamic response model for mildly separated flow.

The network training sets are composed of the nine characteristic angle of attack motions (sinusoidal, ramp-up, and ramp-down) and respective unsteady normal force coefficient response histories, $C_N(t)$, corresponding to one of three freestream Mach numbers, as presented in Table 4.1. The confirmation that each training case is in the range of weakly non-linear behaviour of mildly separated flowfields is presented in Appendix A in Figures A.2 to A.10. The presentation of each training set comprises normalised values with respect to the maximum incidence prior to training. To guarantee adequate representation of the input motion histories and output aerodynamic responses the sample interval in the discrete-time model is 0.01 s.

For the maximum complexity FIR neural network in the population: 2 hidden layers and 10 neurons per hidden layer are used. A maximum time-delay per connection (τ_{max}) of 3 is assumed. The training process has been carried out in 50,000 generations, since the error goal was not achieved. In Table 4.4, the complete set of training parameters is presented.

Training parameters	Value
population size	10
number of crossover points	5
P_t	0.3%
P_n	0.1%
scaling factor for selection	2.0
perturbation constant, β	0.008
number of cycles in updating weight/bias	5
number of steps before forced mutation	1000

Table 4.4: Training parameters for the identification of weakly non-linear unsteady aerodynamic response in mildly separated flow.

Figures 4.4 to 4.6 present a comparison between the desired normal force coefficient response and the FIR neural network output for each of the training sets after the end of the identification process.

The form of the final adapted FIR neural network (including the number of time-delays per connection) is illustrated in Figure 4.7. The respective weight and bias values are presented in Tables 4.5 and 4.6. In Table 4.5, the rows are related to the outputs of each neuron in the FIR neural network (*cf.* Figure 4.7), while the columns are related to the inputs. The row values in parenthesis represent time-delay values. Therefore, each weight value for the respective time-delay (in parenthesis) belongs to the connection linking the neuron number in the row to the neuron number in the column.

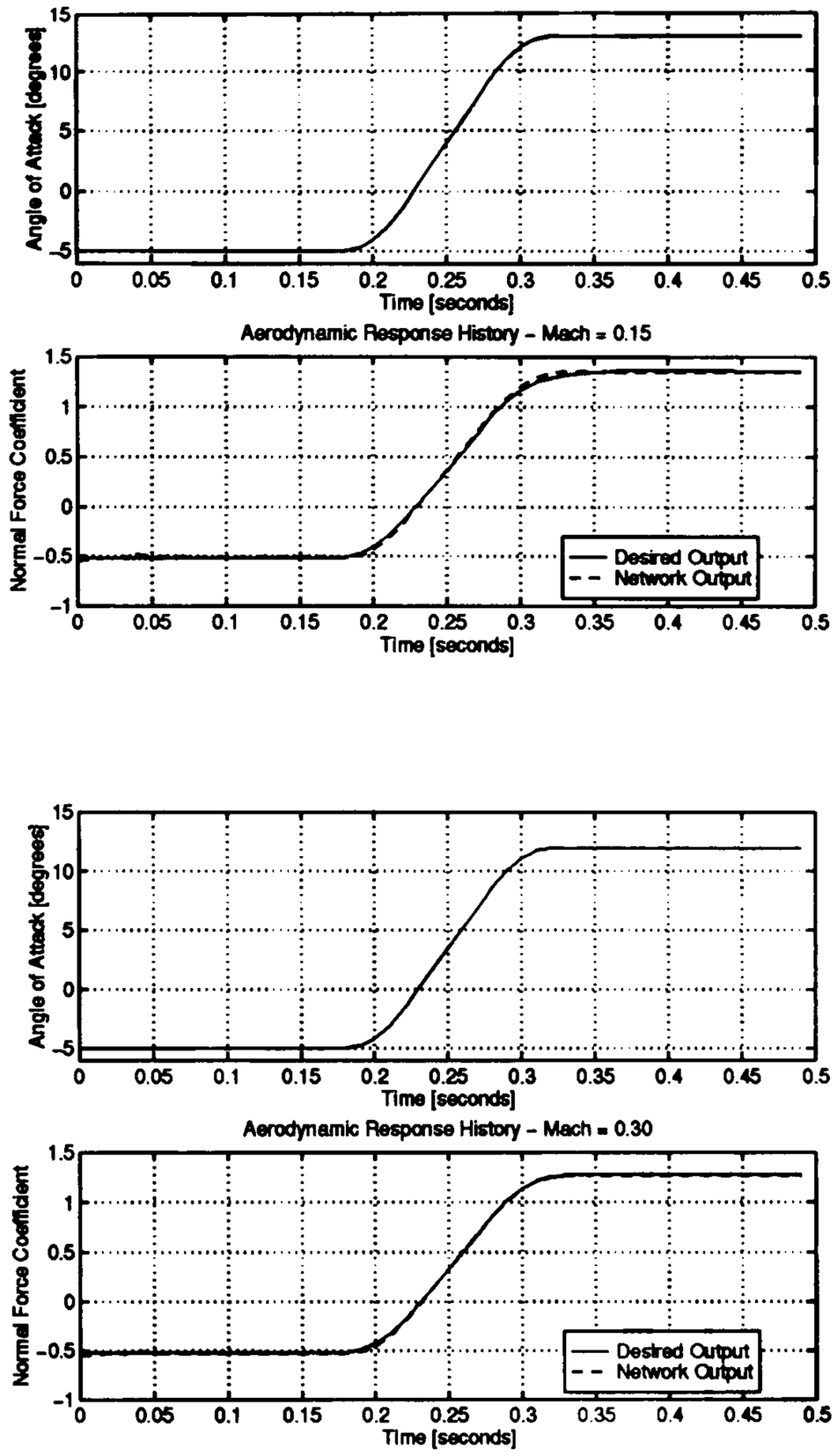


Figure 4.4: Identification of unsteady aerodynamic response in mildly separated flow: Training sets and adapted FIR neural network outputs for the ramp-up cases.

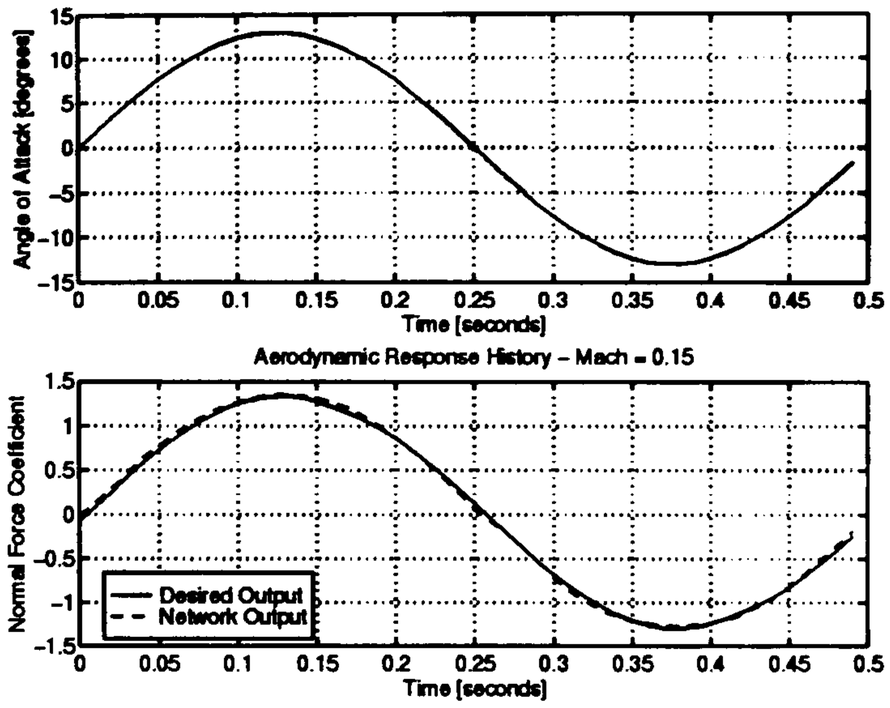


Figure 4.5: Identification of unsteady aerodynamic response in mildly separated flow: Training sets and adapted FIR neural network outputs for the sinusoidal cases.

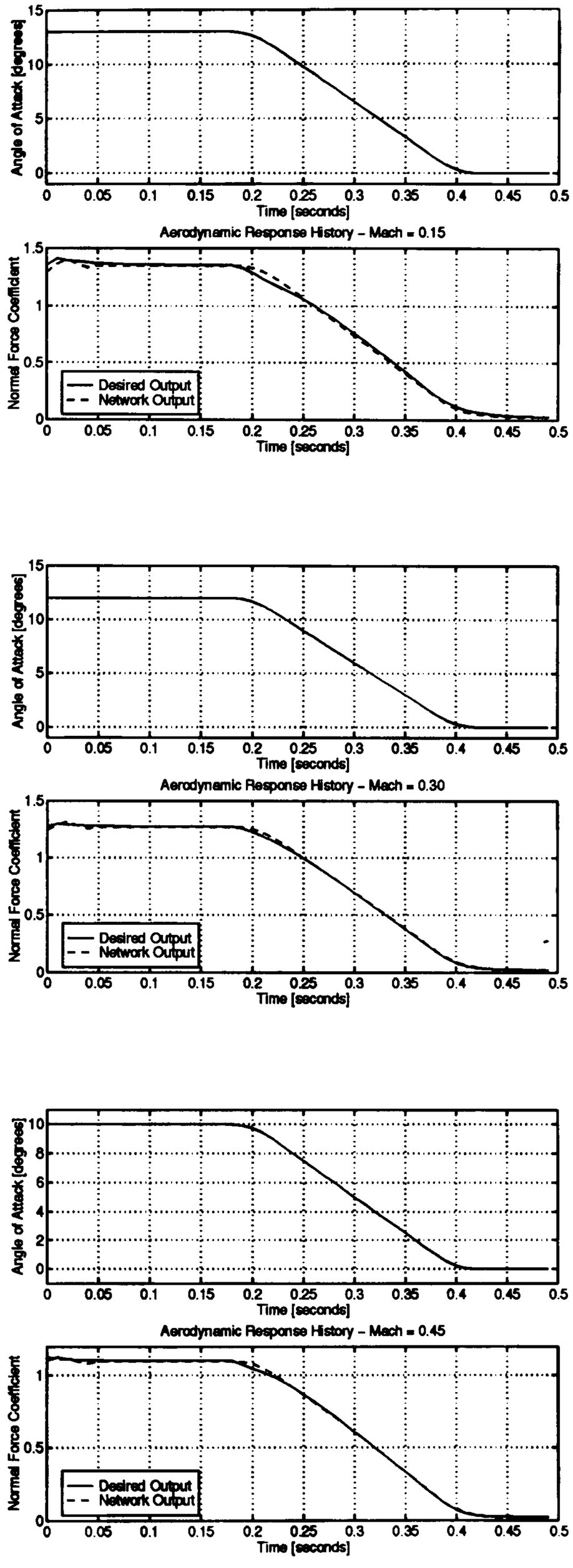


Figure 4.6: Identification of unsteady aerodynamic response in mildly separated flow: Training sets and adapted FIR neural network outputs for the ramp-down cases.

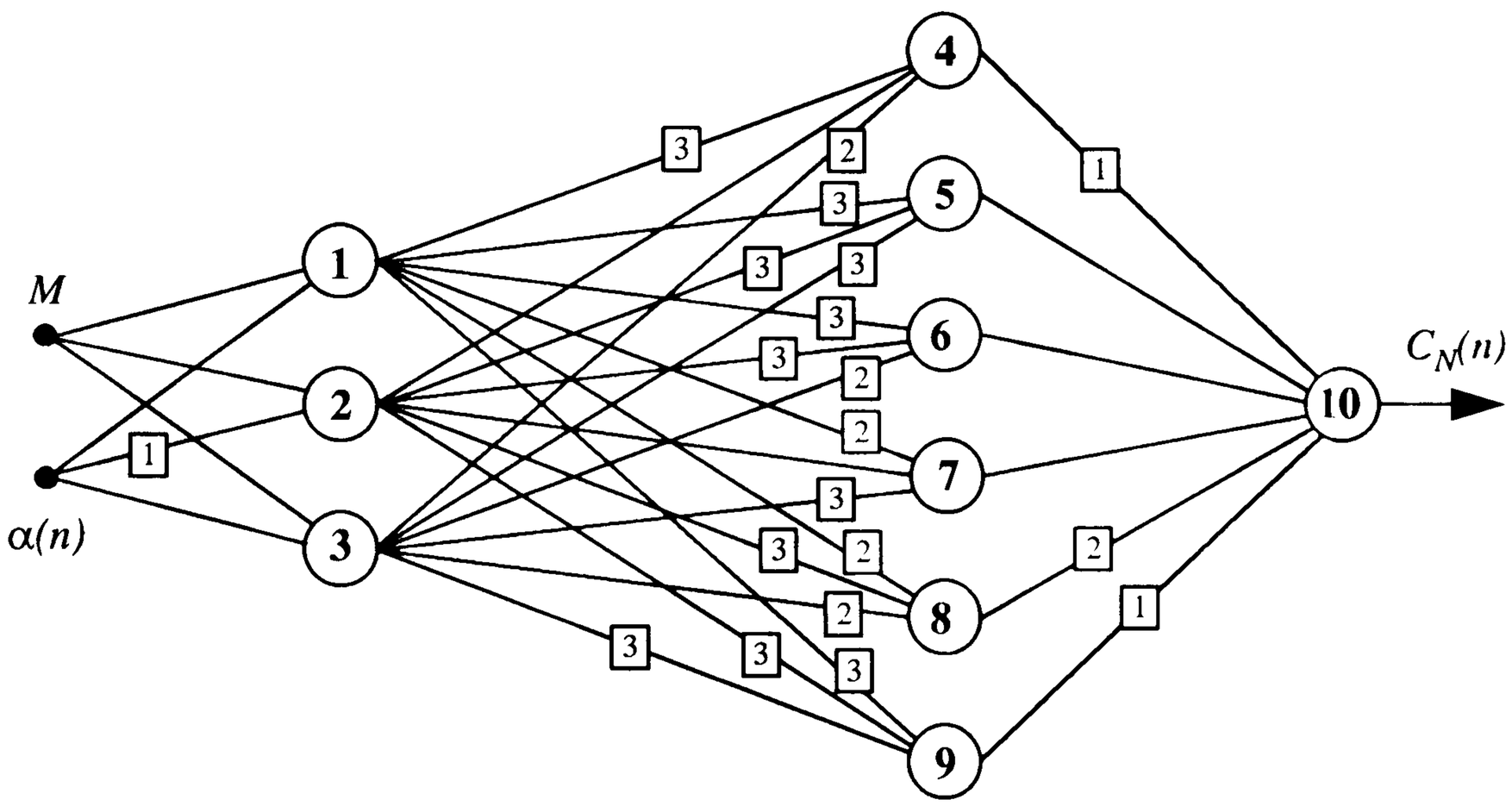


Figure 4.7: Identified FIR network model for the unsteady aerodynamic response in mildly separated flow.

		1	2	3	4	5	6	7	8	9	10
M		0.131	0.144	0.109							
α		-0.110	-0.237	0.120							
	(1)	-	-0.052	-							
	(2)	-	-	-							
	(3)	-	-	-							
1					0.699	-0.860	-0.005	-0.626	0.844	-0.446	
	(1)				-0.319	0.136	0.821	0.360	0.003	-0.215	
	(2)				-0.248	-0.163	0.698	0.009	-0.001	0.204	
	(3)				-0.049	0.325	0.279	-	-	0.315	
2					-0.982	0.142	0.751	0.279	0.532	-0.455	
	(1)				-	0.479	0.504	-	-0.191	-0.904	
	(2)				-	0.005	0.806	-	0.066	0.563	
	(3)				-	-0.396	-0.527	-	0.181	0.950	
3					-0.787	-1.148	-0.795	-0.752	-0.624	-0.306	
	(1)				-0.800	0.208	0.238	-0.391	0.691	0.207	
	(2)				-0.113	0.164	-0.898	0.119	0.569	0.526	
	(3)				-	0.003	-	-0.413	-	-0.027	
4											-0.255
	(1)										-0.661
	(2)										-
	(3)										-
5											-0.151
	(1)										-
	(2)										-
	(3)										-
6											-0.014
	(1)										-
	(2)										-
	(3)										-
7											-0.461
	(1)										-
	(2)										-
	(3)										-
8											-0.773
	(1)										-0.206
	(2)										-0.165
	(3)										-
9											-0.015
	(1)										-0.001
	(2)										-
	(3)										-

Table 4.5: Weight values in the identified FIR network model for the unsteady aerodynamic response in mildly separated flow.

	1	2	3	4	5	6	7	8	9	10
θ	-0.965	0.121	-0.806	0.689	-0.071	-0.823	-0.243	0.049	-0.114	0.480

Table 4.6: Bias values of each neuron of the identified FIR network model of unsteady aerodynamic response in mildly separated flow.

Figure 4.8 shows the convergence characteristics of the algorithm during network training. The sum of squared errors between the Beddoes model and FIR neural network outputs corresponding to the best individual in the population for steps of ten generations is depicted. The error values are normalised with respect to the initial error value, which is determined by the inverse fitness value of the best individual in the population.

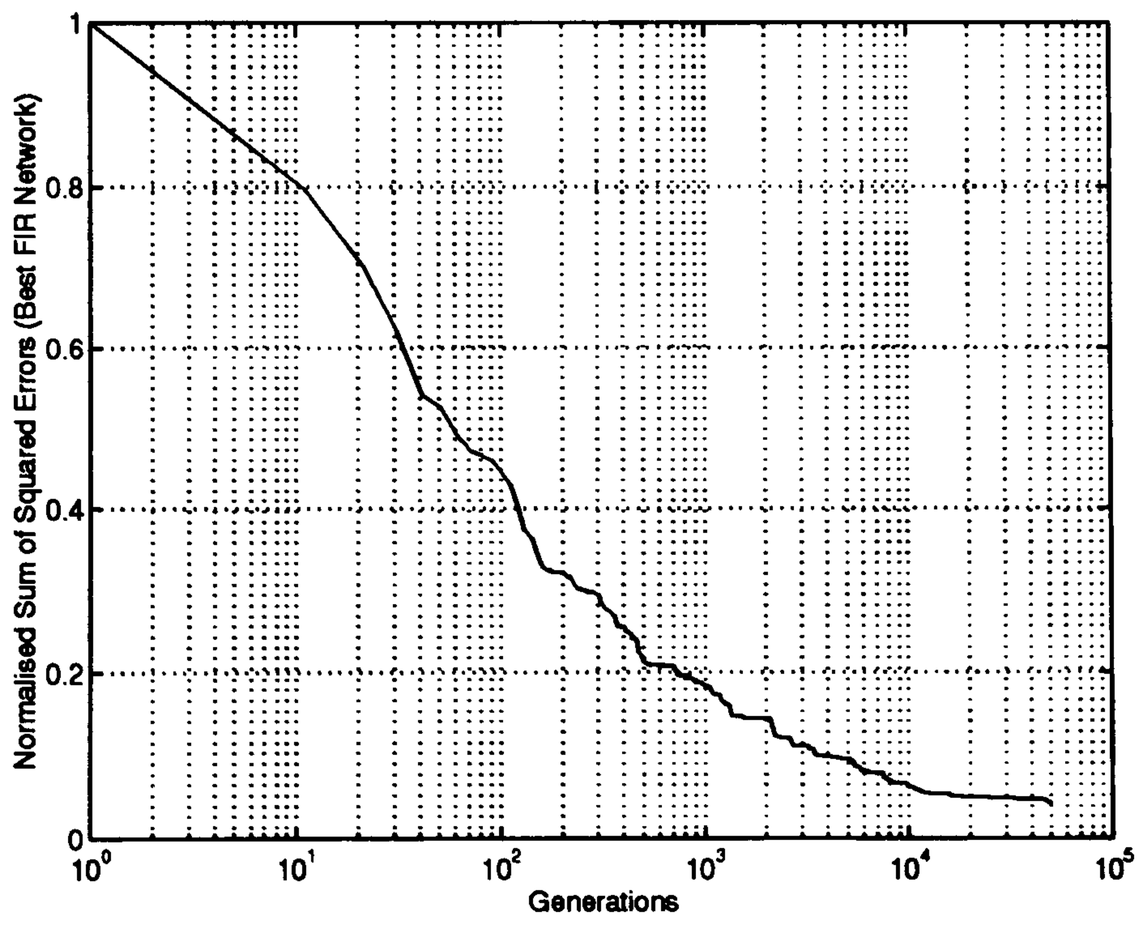


Figure 4.8: Error decay during the identification of the FIR network model of unsteady aerodynamic response in mildly separated flow.

The robustness of the adapted network identified from the prescribed training sets is examined in Figures 4.9 to 4.16. For each case, pitch incidence history and Mach number values different from the ones used to identify the FIR neural network, are used. In Appendix A, in Figures A.11 to A.18, the progression of the separation point for each test case helps to ensure that the correct aspects of the physical phenomena are present in the generalisation tests. The identified FIR neural network model output is then compared to the desired normal force coefficient response. All test cases show satisfactory correlation between the normal force coefficient identified by the FIR neural network model and the respective Beddoes model outputs.

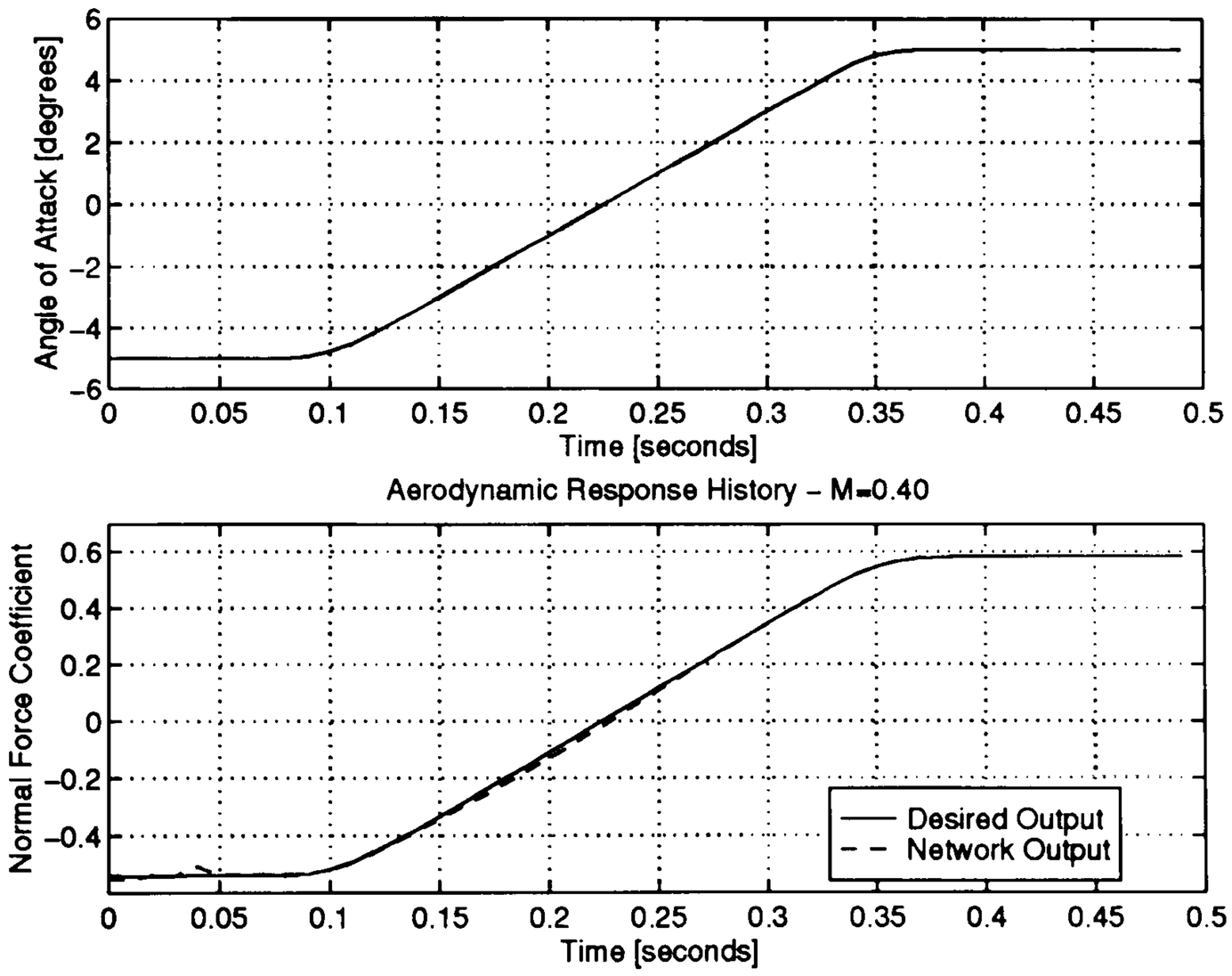


Figure 4.9: FIR network model of mildly separated unsteady aerodynamic response to arbitrary motion histories in the linear range.

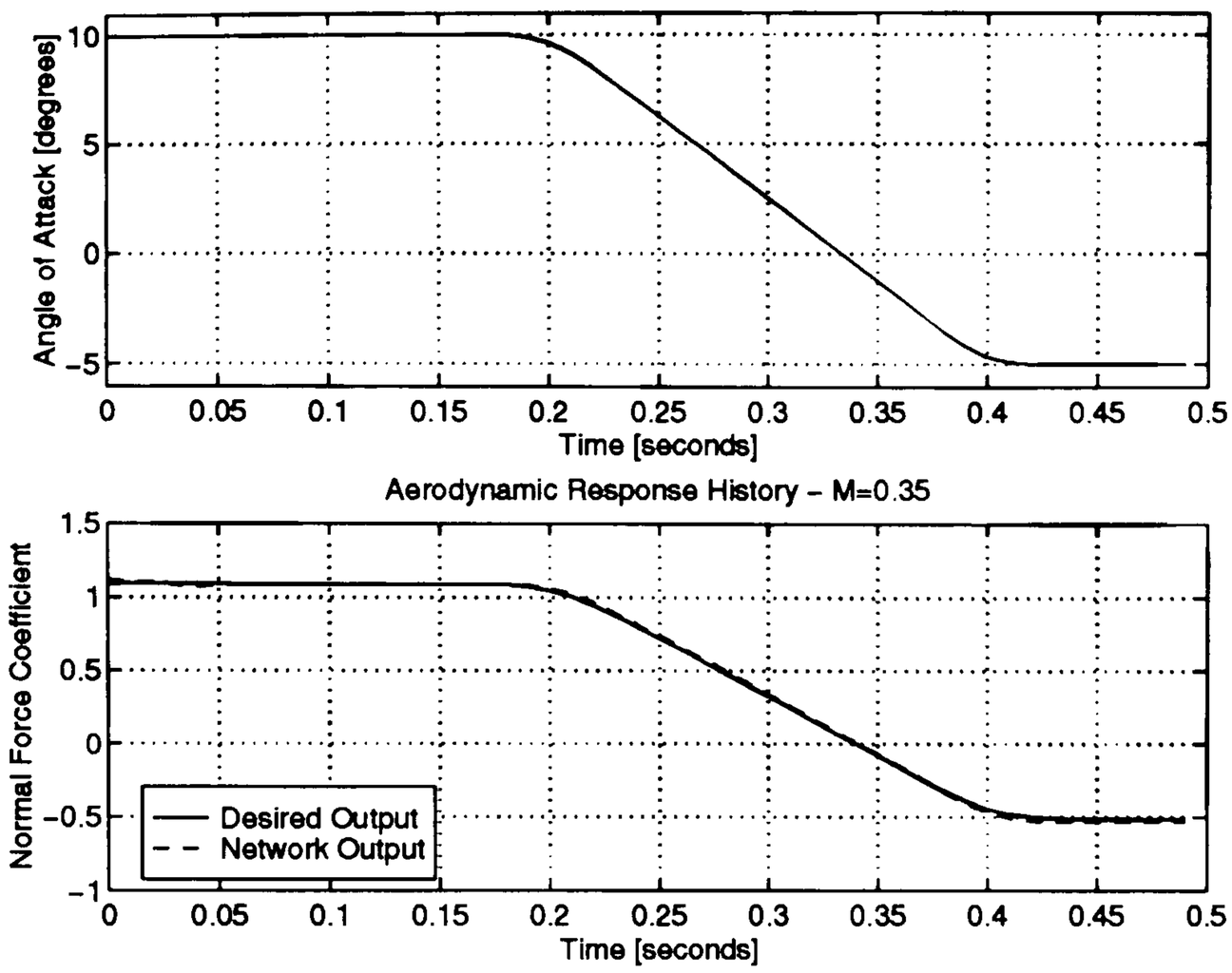


Figure 4.10: FIR network model of mildly separated unsteady aerodynamic response to arbitrary motion history and Mach number.

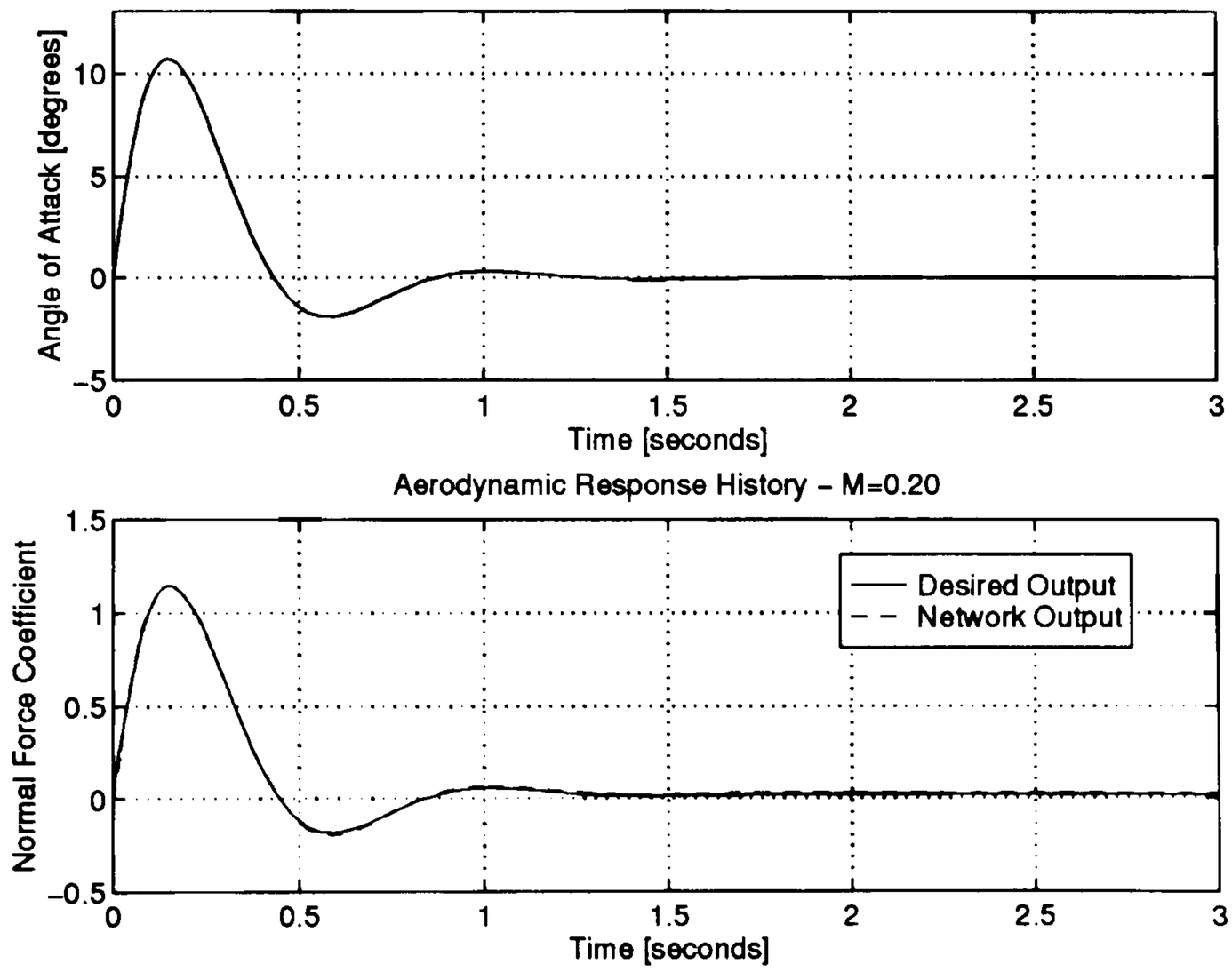


Figure 4.11: FIR network model of mildly separated unsteady aerodynamic response to arbitrary motion history and Mach number.

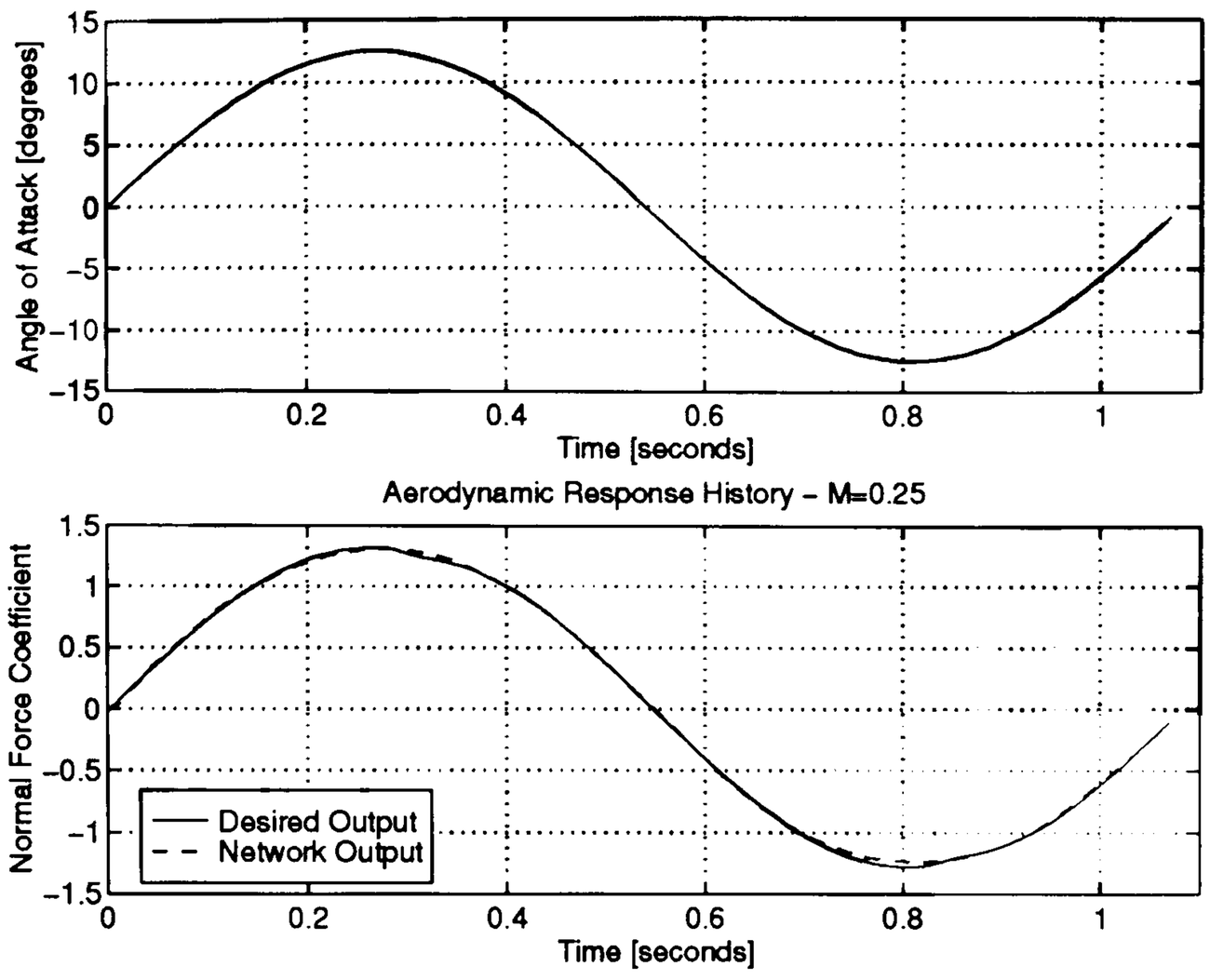


Figure 4.12: FIR network model of mildly separated unsteady aerodynamic response to arbitrary motion histories at lower frequency than that of the training sets.

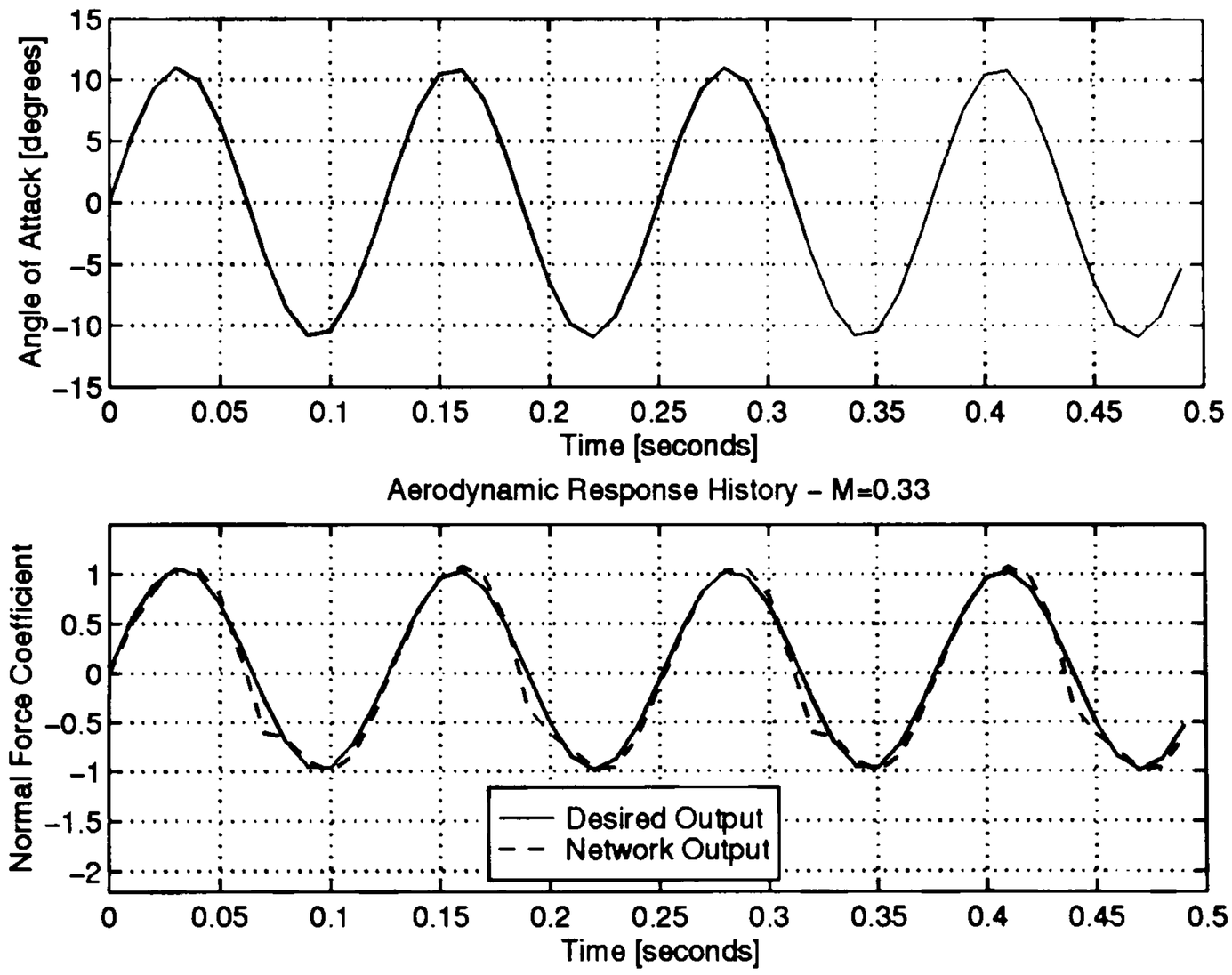


Figure 4.13: FIR network model of mildly separated unsteady aerodynamic response to arbitrary motion histories at higher frequency than that of the training sets.

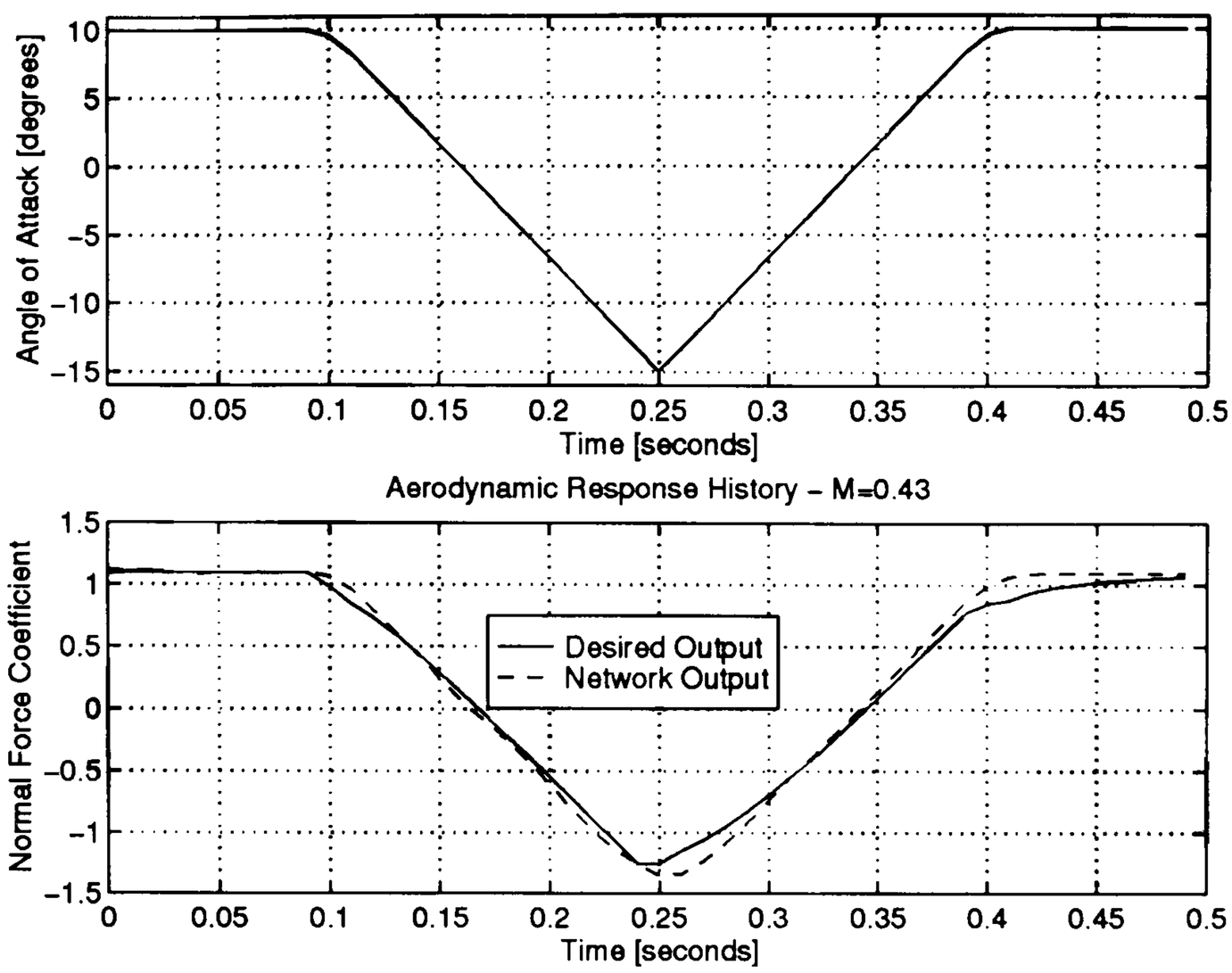


Figure 4.14: FIR network model of mildly separated unsteady aerodynamic response to severe arbitrary motion history and Mach number.

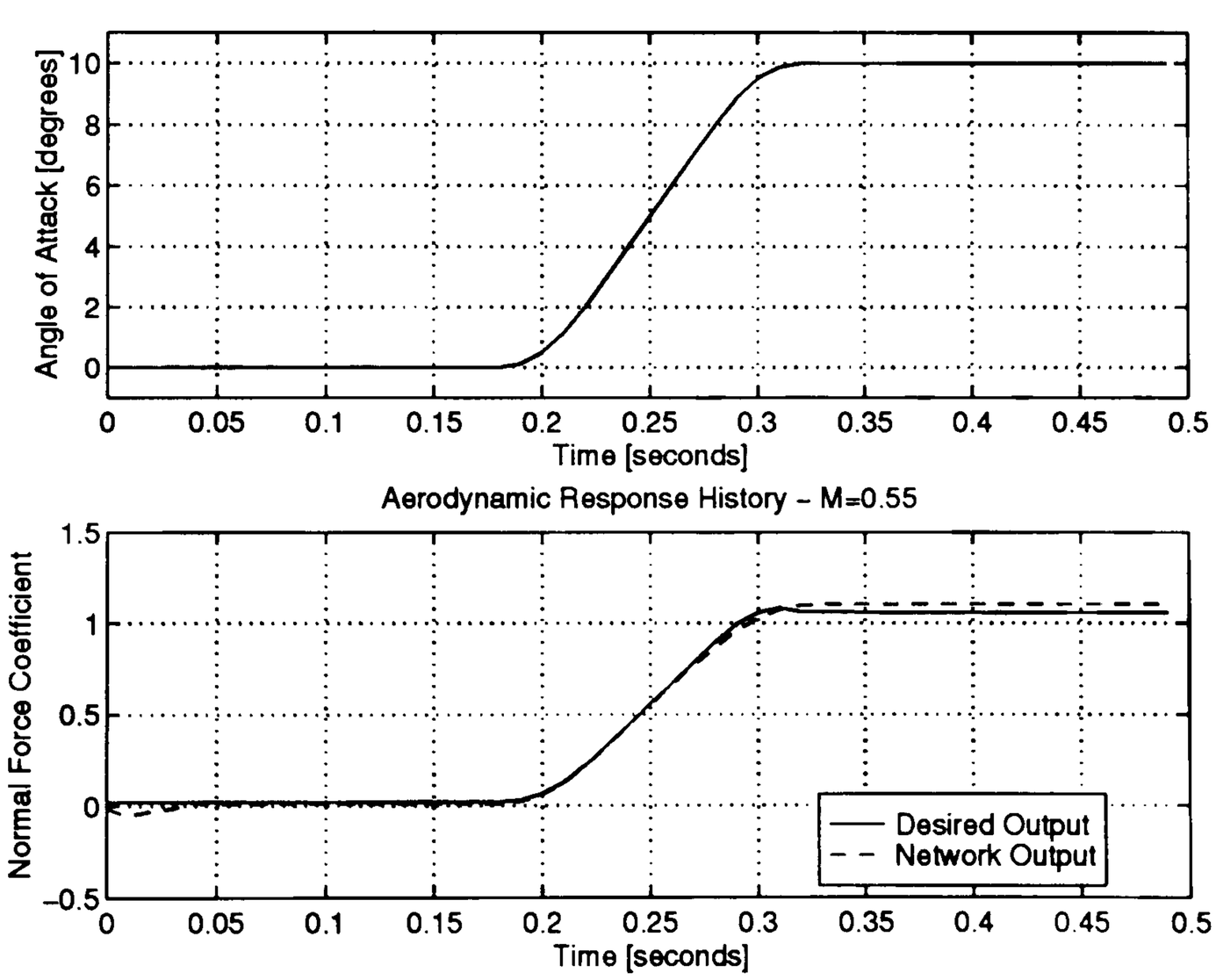


Figure 4.15: FIR network model of mildly separated unsteady aerodynamic response to arbitrary motion history and Mach number beyond training range.

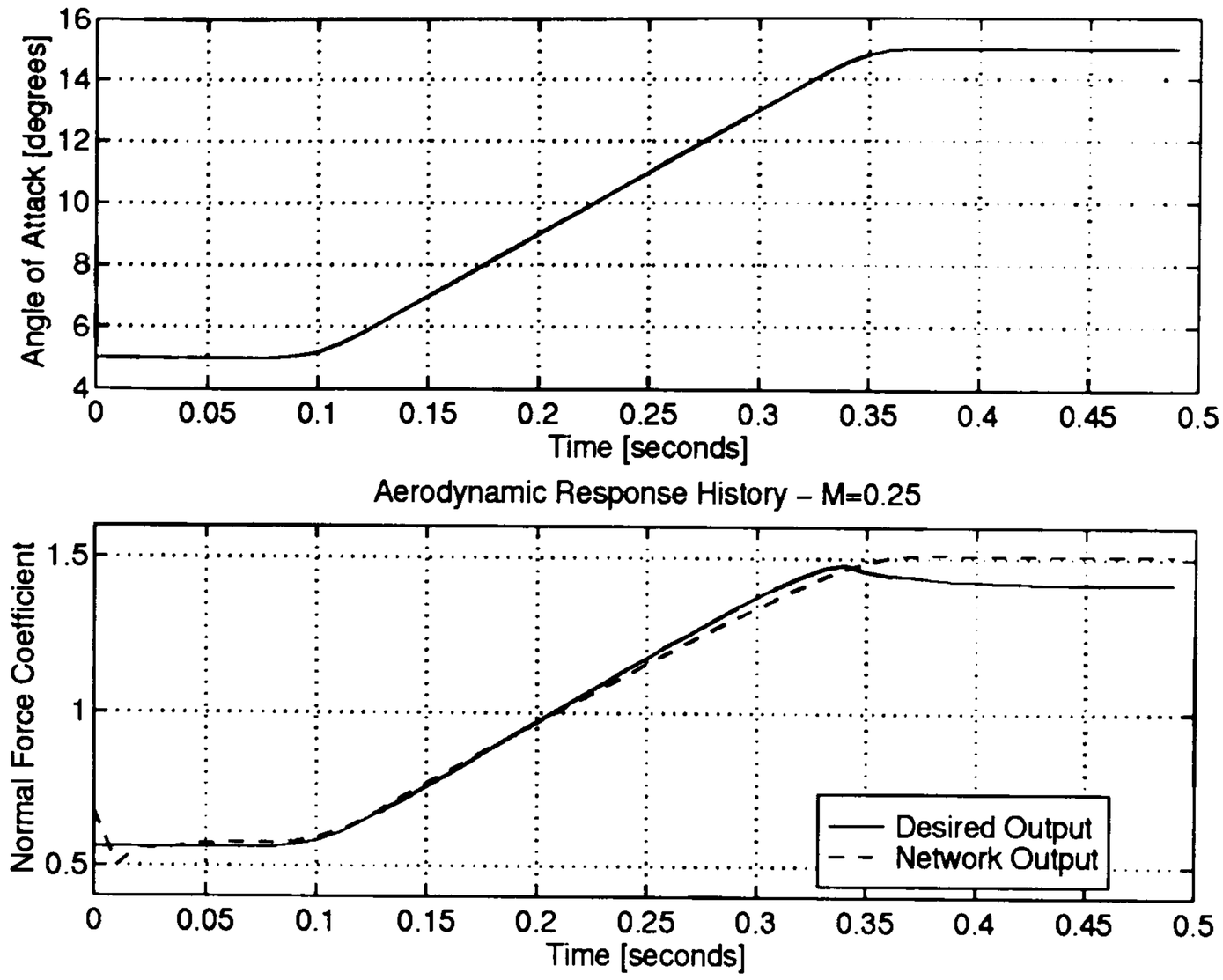


Figure 4.16: FIR network model of mildly separated unsteady aerodynamic response to arbitrary motion histories beyond training range.

4.4 Approximation of the Unsteady Aerodynamic Response in Transonic Flow

Multi-layer functional models of unsteady transonic aerodynamic response are identified for two different problems. The first one considers the identification of a FIR network model for the unsteady transonic aerodynamic response to variations in the incidence angle at a fixed Mach number ($M = 0.65$), while the second problem considers the model identification for a range of Mach numbers. Both problems are based on the features shown in Figure 4.17. for a freestream sonic velocity of 340.5 $m.s^{-1}$, and the airfoil chord length of 0.55 m.

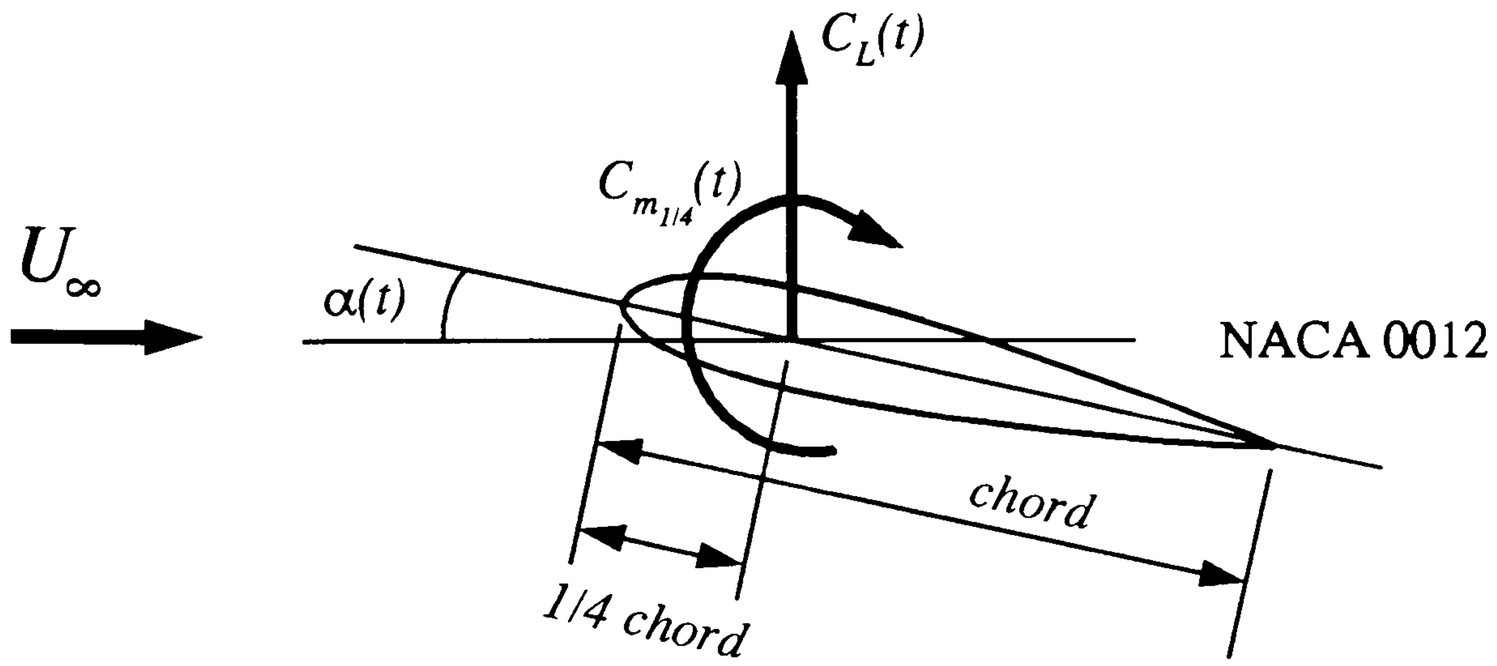


Figure 4.17: Parameters for the identification of the FIR neural network model of the unsteady aerodynamic response in the transonic regime.

In comparison with the problem presented in the previous section, here the complexity of the input-output mappings naturally demands more generations for training, as well as different training parameters.

4.4.1 Unsteady Aerodynamic Response in Transonic Flow at Fixed Mach Number

In the present case, the training algorithm is used to identify the non-linear functional relationship between both, lift force coefficient, $C_L(t)$, pitch moment coefficient at 25% chord length, $C_{m_{1/4}}(t)$, and incidence histories, α_t, for a NACA 0012 airfoil operating at a fixed value of Mach number ($M = 0.65$). Equation 4.4 describes the functional representation adopted; that is,

$$\left\{ \begin{array}{c} C_L(t) \\ C_{m_{1/4}}(t) \end{array} \right\} = \mathcal{MF}_M[\alpha_t] \tag{4.4}$$

The aerodynamic response to the characteristic motions used for training are obtained from the CFD code described in Section 4.2.2. Table 4.7 presents a description of each one of the training sets used in the identification process. For

all training cases, the appearance and dynamic motion of shock waves responsible for non-linear behaviour of the unsteady aerodynamic response can be observed in Appendix B. The three characteristic motions for training have a sample interval of 0.002 s, to adequate the representation of the input motion histories and output aerodynamic responses.

Characteristic motion	Range
sinusoidal	$\alpha_{mean} = 0°$; amplitude= 4°, frequency 10 Hz
sigmoidal ramp-up	$\alpha_{min} = -1°$; $\alpha_{max} = 4°$
pulse-down	$\alpha_{initial} = 1°$; $\alpha_{pulse} = -4.5°$; $\alpha_{final} = 1°$

Table 4.7: Training set motions for the identification of the unsteady aerodynamic response model in the transonic regime (fixed $M = 0.65$).

In order to help the convergence, the training process is carried out in stages, with some training parameters altered from one stage to another. A trial-and-error approach is used to obtain new training values for each stage. Here, a three stage training process, totalising 250,000 generations, has been carried out. For the maximum complexity FIR network architecture in the population: 2 hidden layers and 10 neurons per hidden layer are used. A maximum time-delay per connection (τ_{max}) of 6 is assumed. Table 4.8 presents the complete set of training parameters.

Training parameters	Value		
	Stage 1	Stage 2	Stage 3
population size	14	14	14
number of crossover points	13	13	13
P_t and P_n	0.5%	0.5%	0.5%
scaling factor for selection	2.0	2.0	2.0
perturbation constant, β	0.0001	0.0001	0.001
number of cycles in updating weight/bias	5	5	5
number of steps before forced mutation	200	100	100
number of generations	100.000	100,000	50,000

Table 4.8: Training parameters for the identification of unsteady aerodynamic response model in the transonic regime (fixed $M = 0.65$).

Figures 4.18 to 4.20 present a comparison between lift force coefficient and pitch moment coefficient responses obtained by the Euler CFD code and the respective FIR neural network outputs for each of the training sets after completion of the identification process.

The architecture and time-delay distribution of the identified FIR network model is presented in Figure 4.21, while the respective values of weight and bias are respectively presented in Tables 4.9 to 4.10.

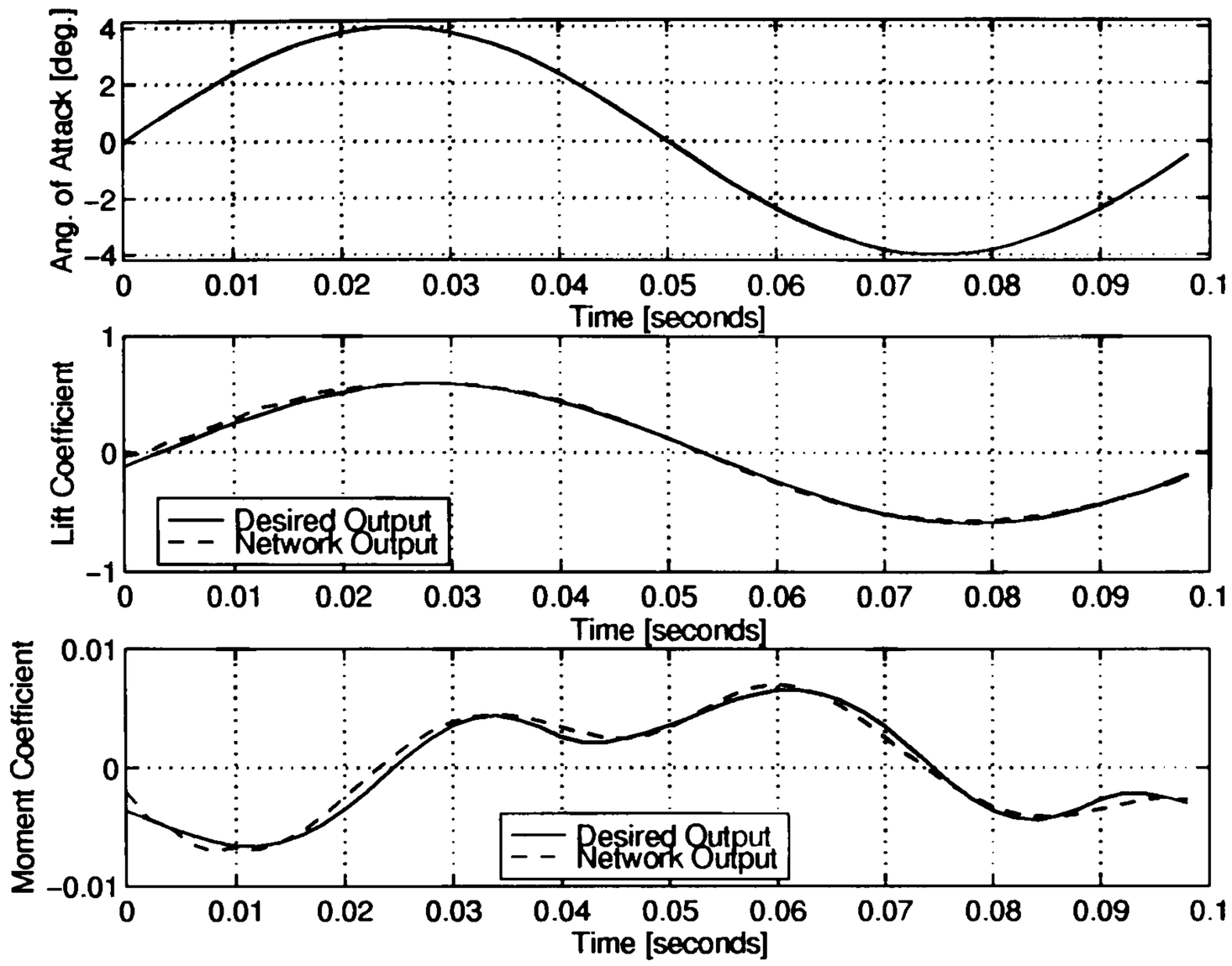


Figure 4.18: Identification of the unsteady aerodynamic response model in the transonic regime (fixed $M = 0.65$): Euler CFD code and FIR network outputs after training for the sinusoidal case.

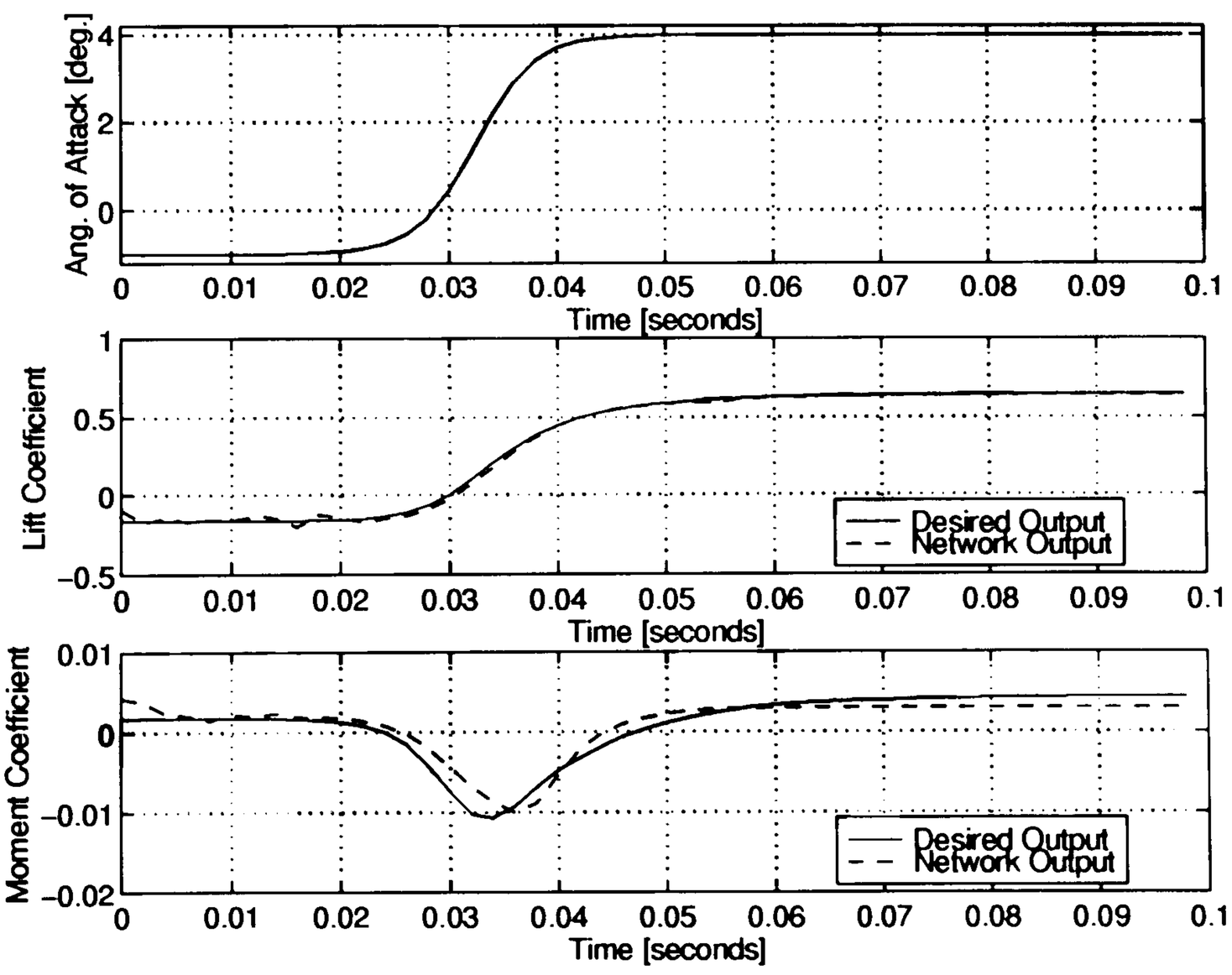


Figure 4.19: Identification of the unsteady aerodynamic response model in the transonic regime (fixed $M = 0.65$): Euler CFD code and FIR network outputs after training for the sigmoidal ramp-up case.

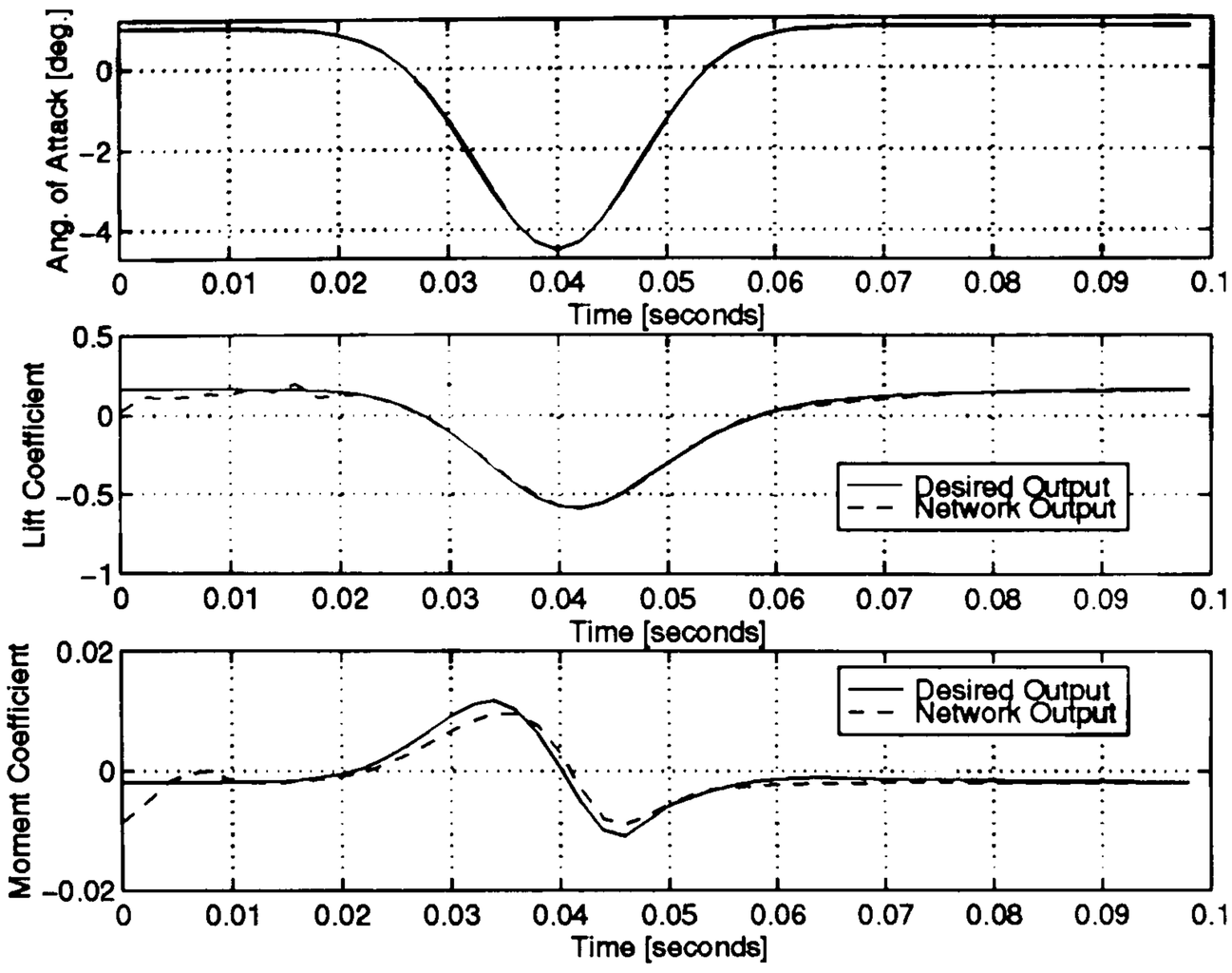


Figure 4.20: Identification of the unsteady aerodynamic response model in the transonic regime (fixed $M = 0.65$): Euler CFD code and FIR network outputs after training for the pulse-down case.

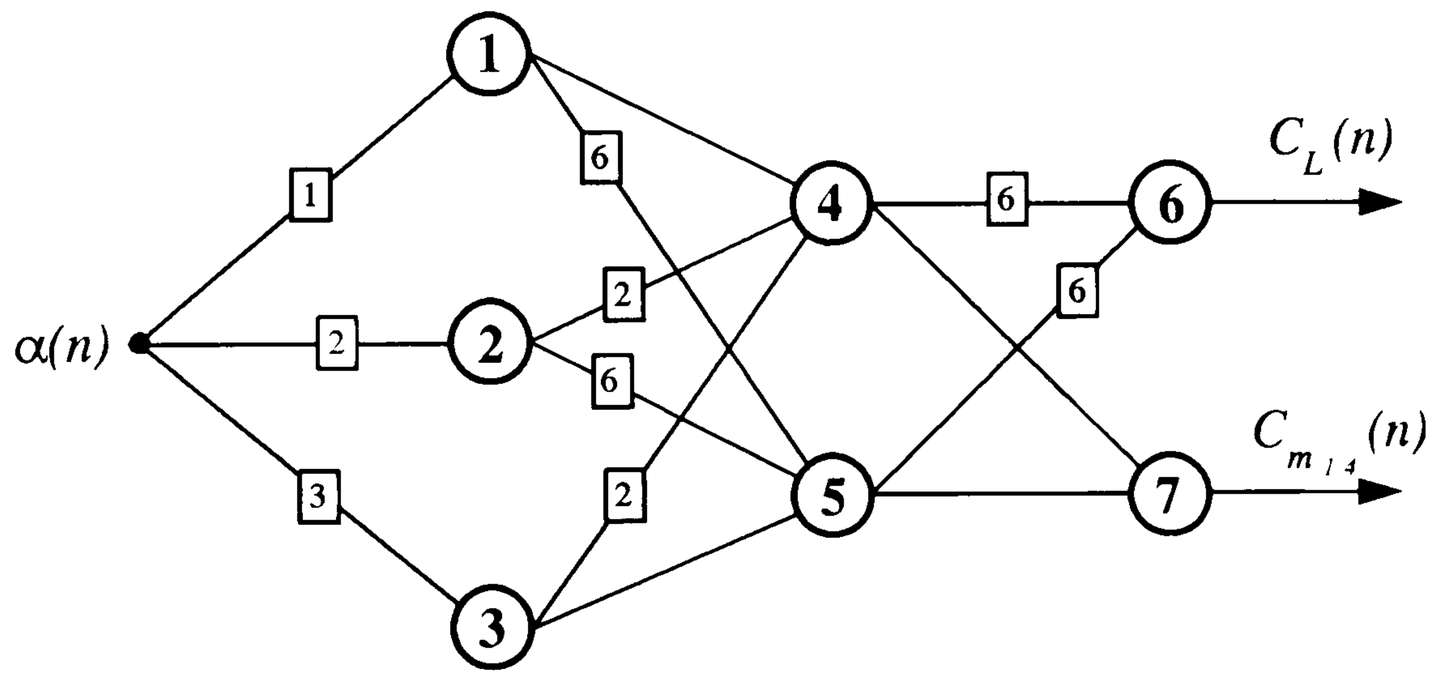


Figure 4.21: Identified FIR neural network model for the unsteady aerodynamic response model in the transonic regime (fixed $M = 0.65$).

		1	2	3	4	5	6	7
α		0.046	-0.323	-0.422				
	(1)	0.060	-0.298	0.191				
	(2)	-	0.634	0.059				
	(3)	-	-	-0.108				
	(4)	-	-	-				
	(5)	-	-	-				
	(6)	-	-	-				
1					-0.362	-0.961		
	(1)				-	0.234		
	(2)				-	-0.243		
	(3)				-	0.099		
	(4)				-	-0.068		
	(5)				-	-0.054		
	(6)				-	-0.053		
2					-0.175	0.185		
	(1)				0.022	0.148		
	(2)				-0.001	0.045		
	(3)				-	-0.160		
	(4)				-	-0.001		
	(5)				-	0.020		
	(6)				-	-0.026		
3					-0.025	0.005		
	(1)				-0.001	-		
	(2)				-0.213	-		
	(3)				-	-		
	(4)				-	-		
	(5)				-	-		
	(6)				-	-		
4						-0.383	-0.241	
	(1)					0.034	-	
	(2)					-0.055	-	
	(3)					0.061	-	
	(4)					-0.012	-	
	(5)					-0.057	-	
	(6)					-0.020	-	
5						-0.686	-0.027	
	(1)					-0.020	-	
	(2)					-0.101	-	
	(3)					-0.691	-	
	(4)					0.385	-	
	(5)					-0.391	-	
	(6)					-0.149	-	

Table 4.9: Weight values in the identified FIR network model for the unsteady aerodynamic response in the transonic regime (fixed $M = 0.65$).

	1	2	3	4	5	6	7
θ	-0.403	-0.090	-0.296	-0.553	-0.141	-0.222	-0.114

Table 4.10: Bias values of each neuron of the identified FIR network model for the unsteady aerodynamic response in the transonic regime (fixed $M = 0.65$).

Figure 4.22 presents the convergence characteristics, in steps of ten generations, of the training process in terms of the normalised sum of squared errors between Euler CFD code and adapted FIR neural network outputs. The best individual in the population with respect to the initial error value, which is determined by the inverse fitness value of the best individual in the population, is considered. The generalisation properties of the identified FIR network model are examined in Figures 4.23 to 4.30, by presenting arbitrary incidence motion histories to the identified model and comparing Euler CFD code and FIR network outputs.

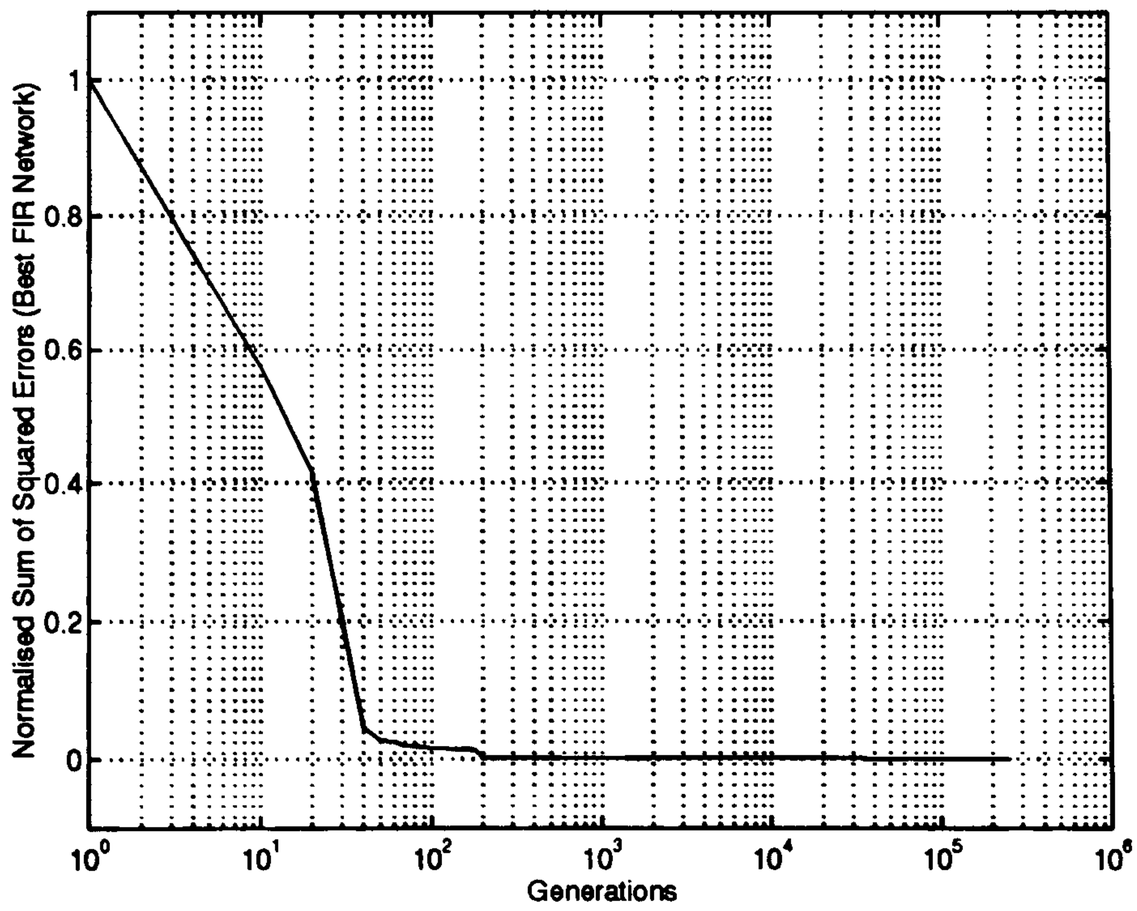


Figure 4.22: Error decay during the identification of the FIR network model of unsteady aerodynamic response in the transonic regime (fixed $M = 0.65$).

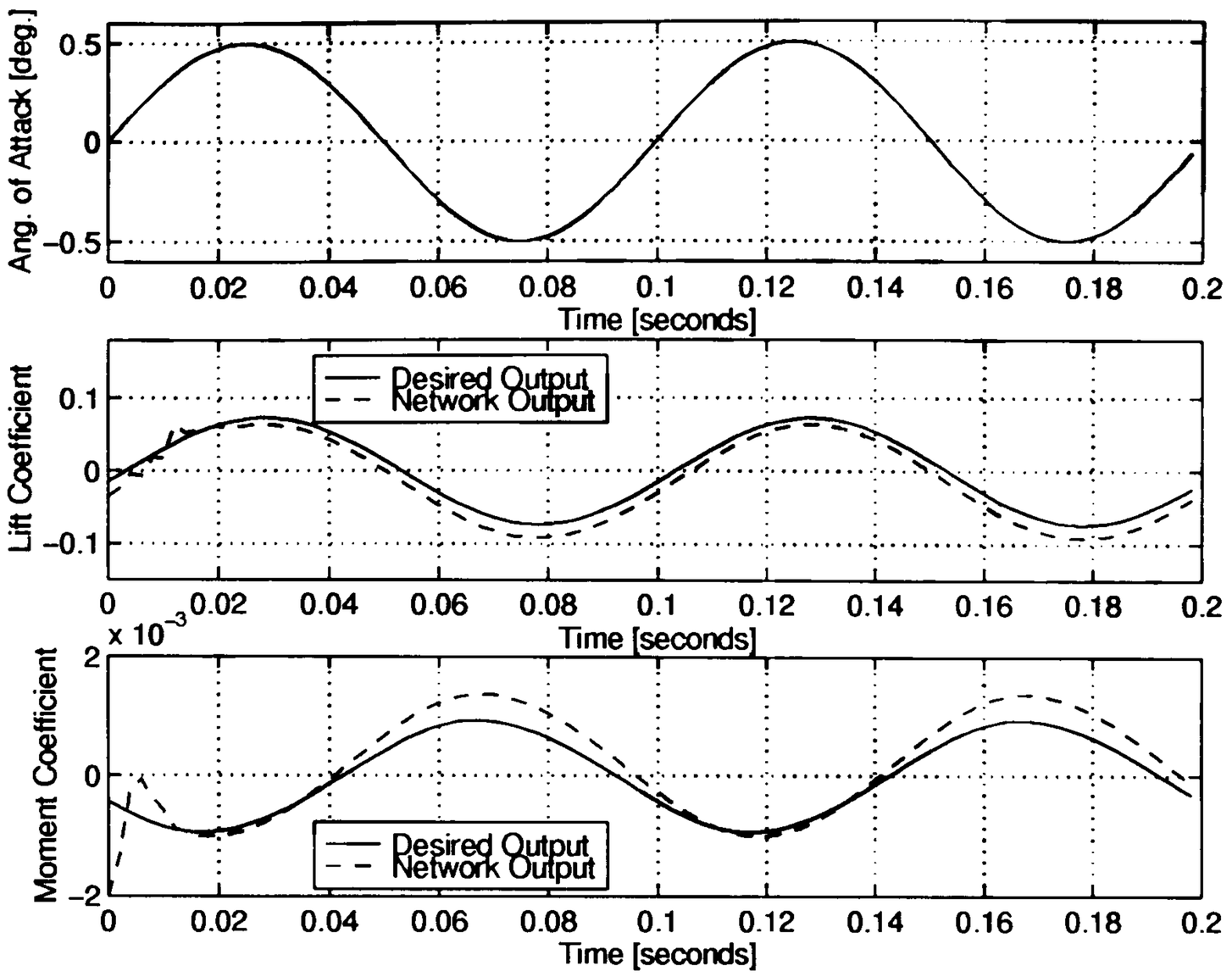


Figure 4.23: FIR network model of unsteady aerodynamic responses in the transonic regime (fixed $M = 0.65$) to arbitrary motion histories in the linear range.

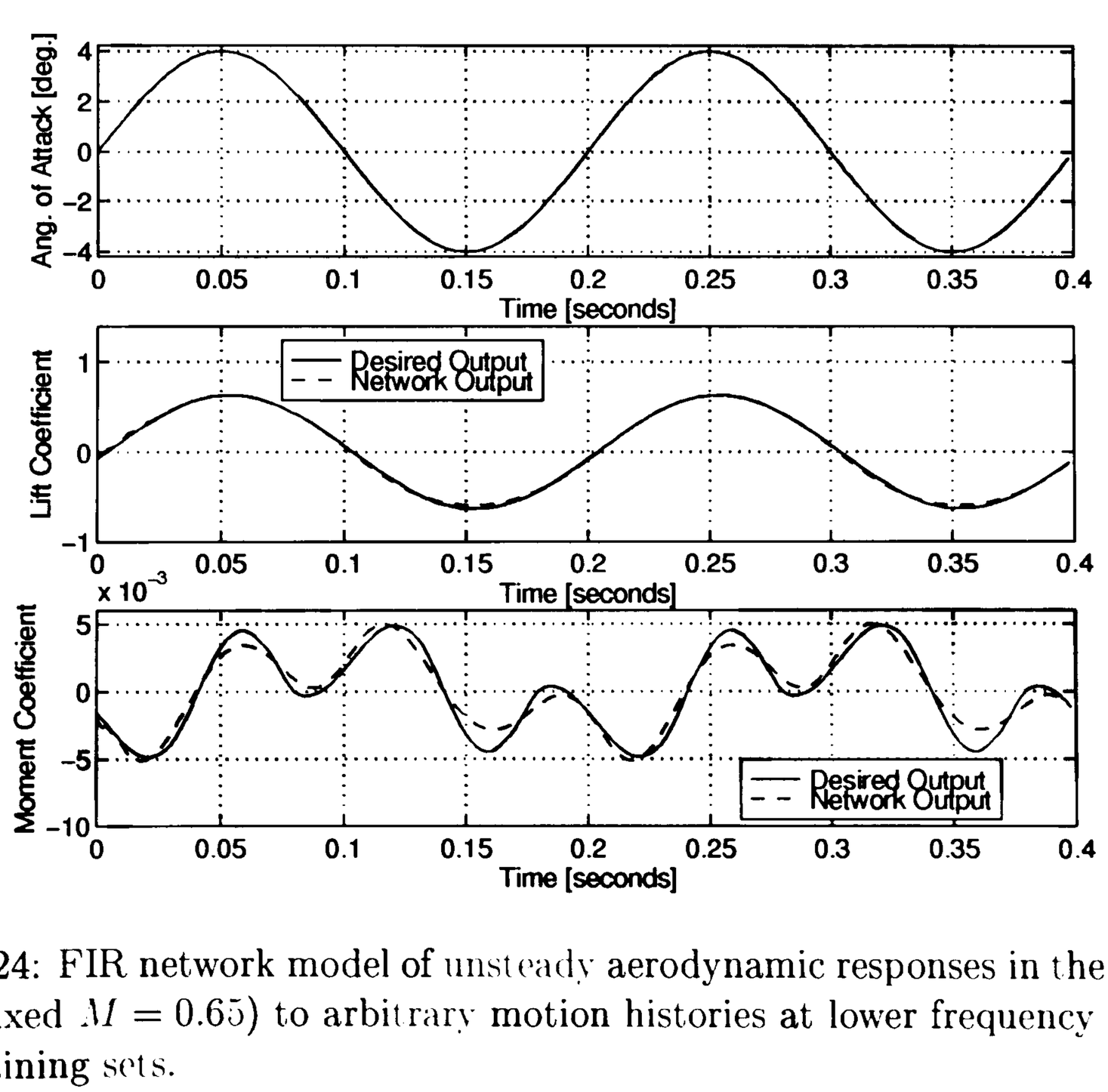


Figure 4.24: FIR network model of unsteady aerodynamic responses in the transonic regime (fixed $M = 0.65$) to arbitrary motion histories at lower frequency than that of the training sets.

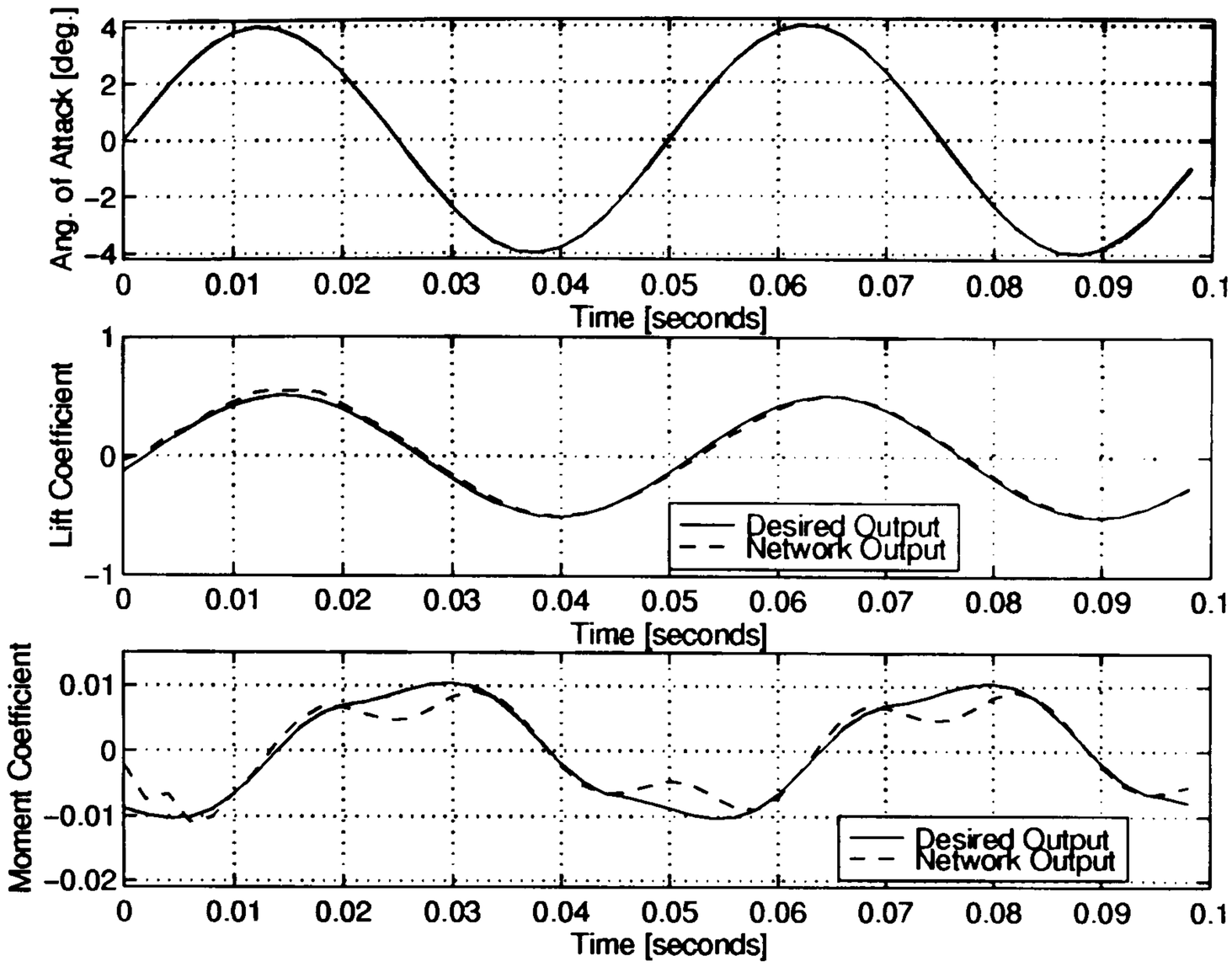


Figure 4.25: FIR network model of unsteady aerodynamic responses in the transonic regime (fixed $M = 0.65$) to arbitrary motion histories at higher frequency than that of the training sets.

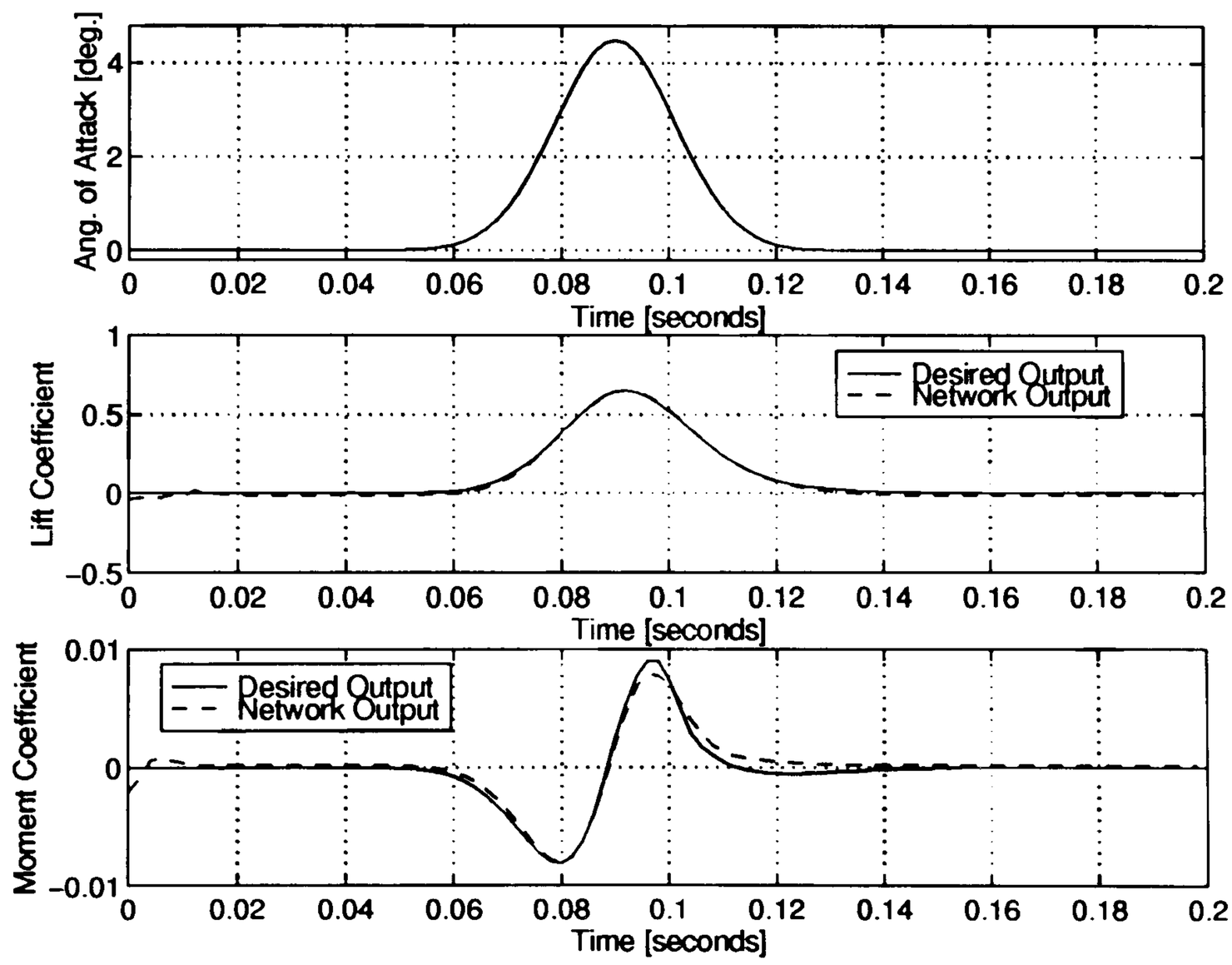


Figure 4.26: FIR network model of unsteady aerodynamic responses in the transonic regime (fixed $M = 0.65$) to arbitrary motion histories.

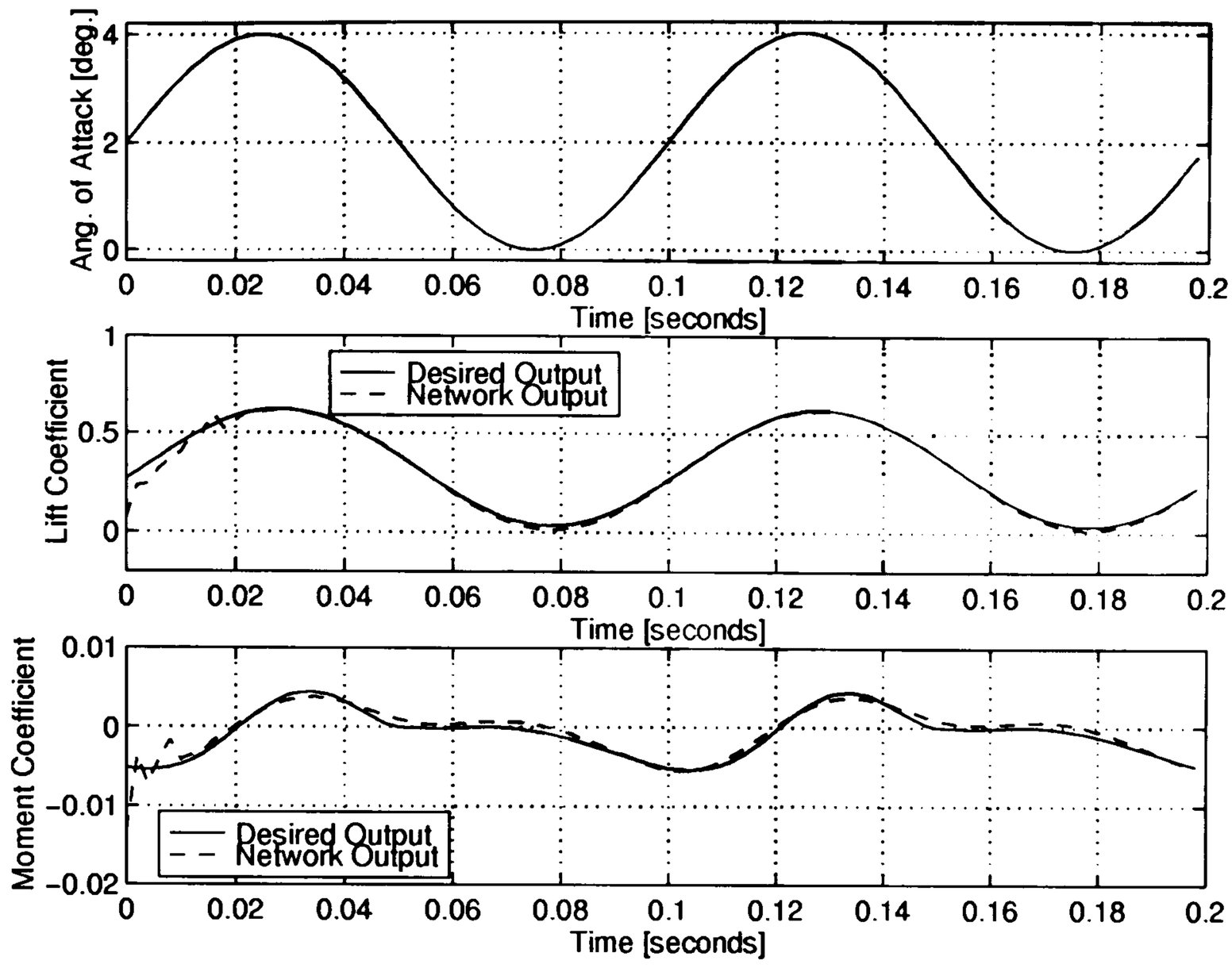


Figure 4.27: FIR network model of unsteady aerodynamic responses in the transonic regime (fixed $M = 0.65$) to arbitrary motion histories.

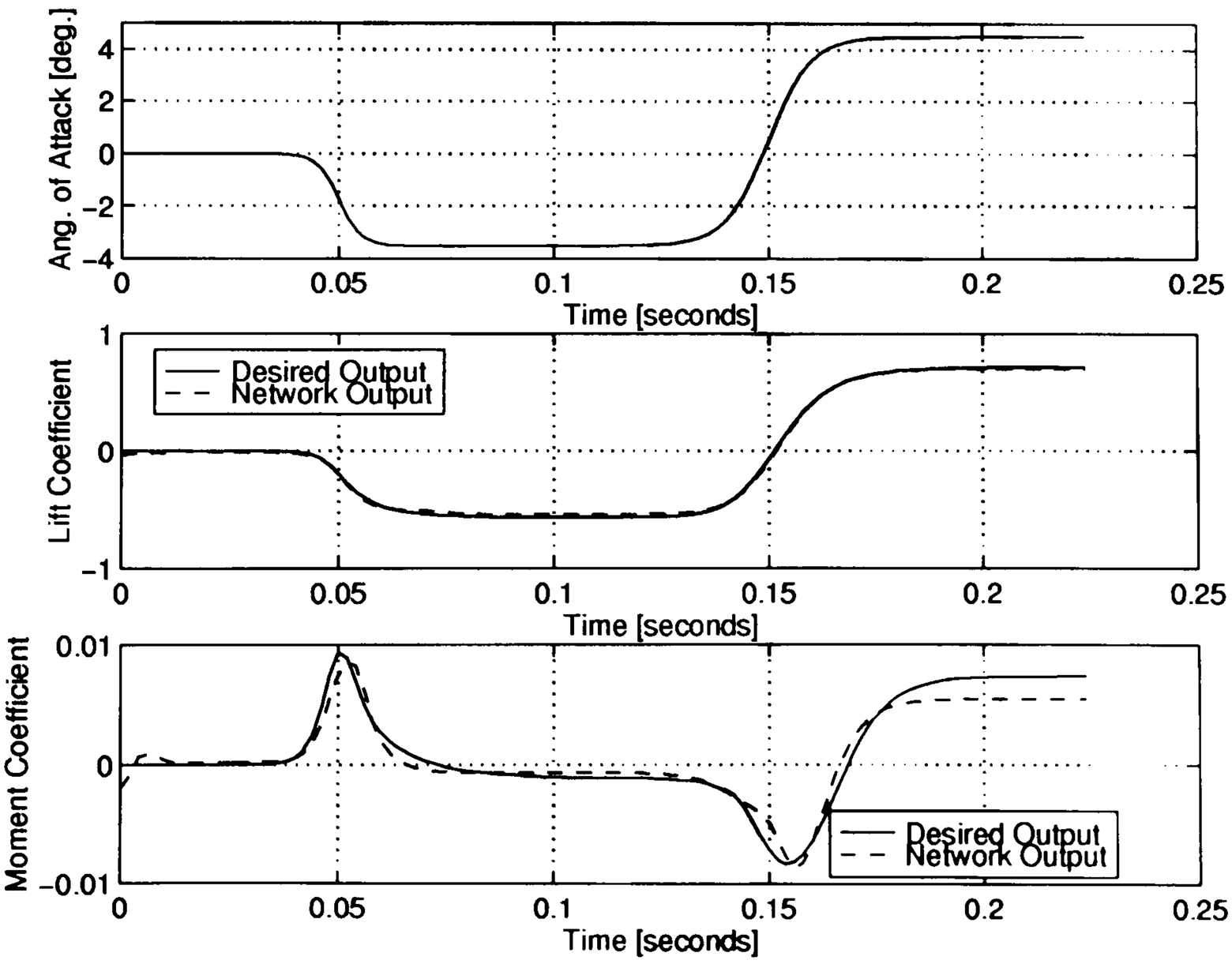


Figure 4.28: FIR network model of unsteady aerodynamic responses in the transonic regime (fixed $M = 0.65$) to arbitrary motion histories.

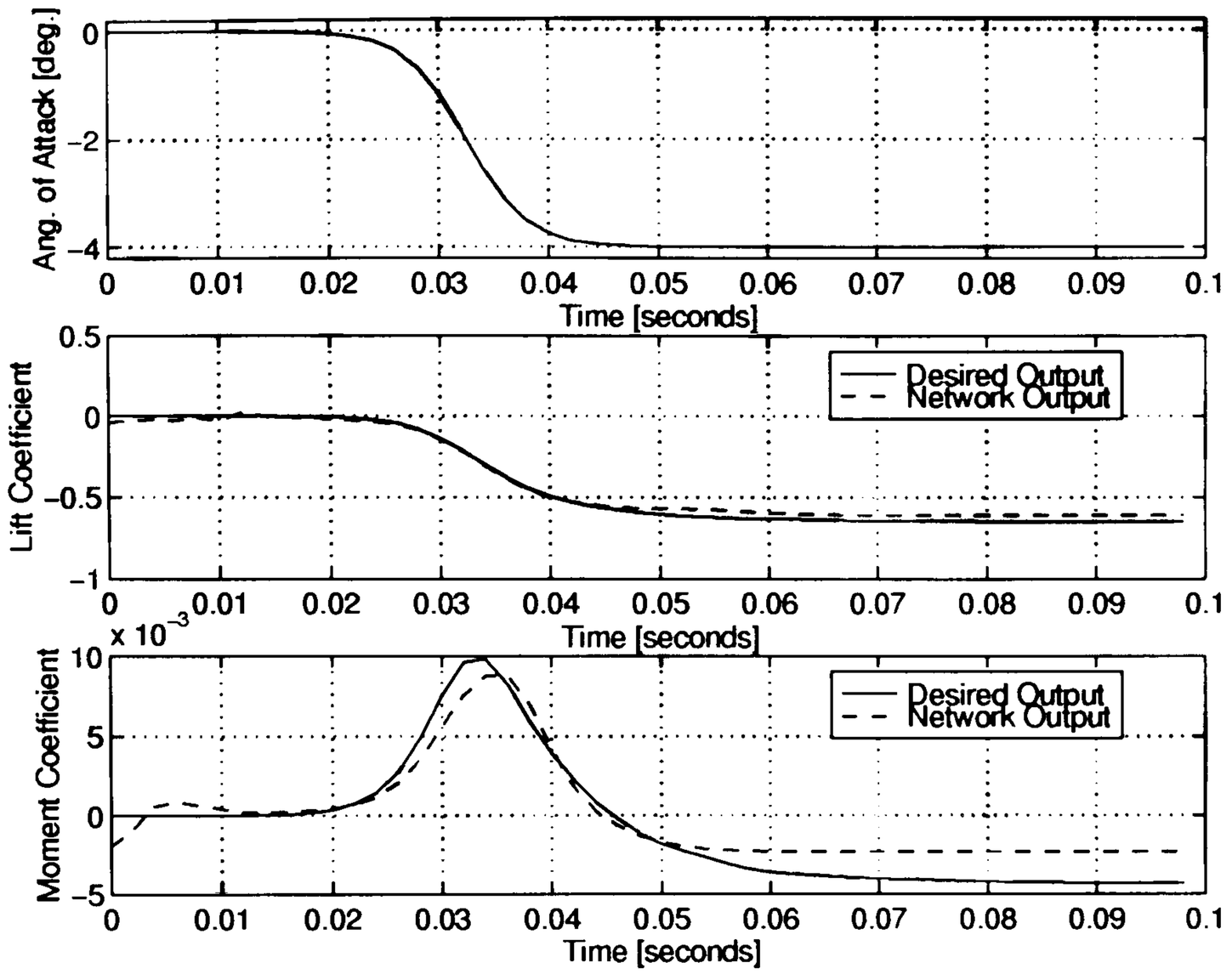


Figure 4.29: FIR network model of unsteady aerodynamic responses in the transonic regime (fixed $M = 0.65$) to arbitrary motion histories.

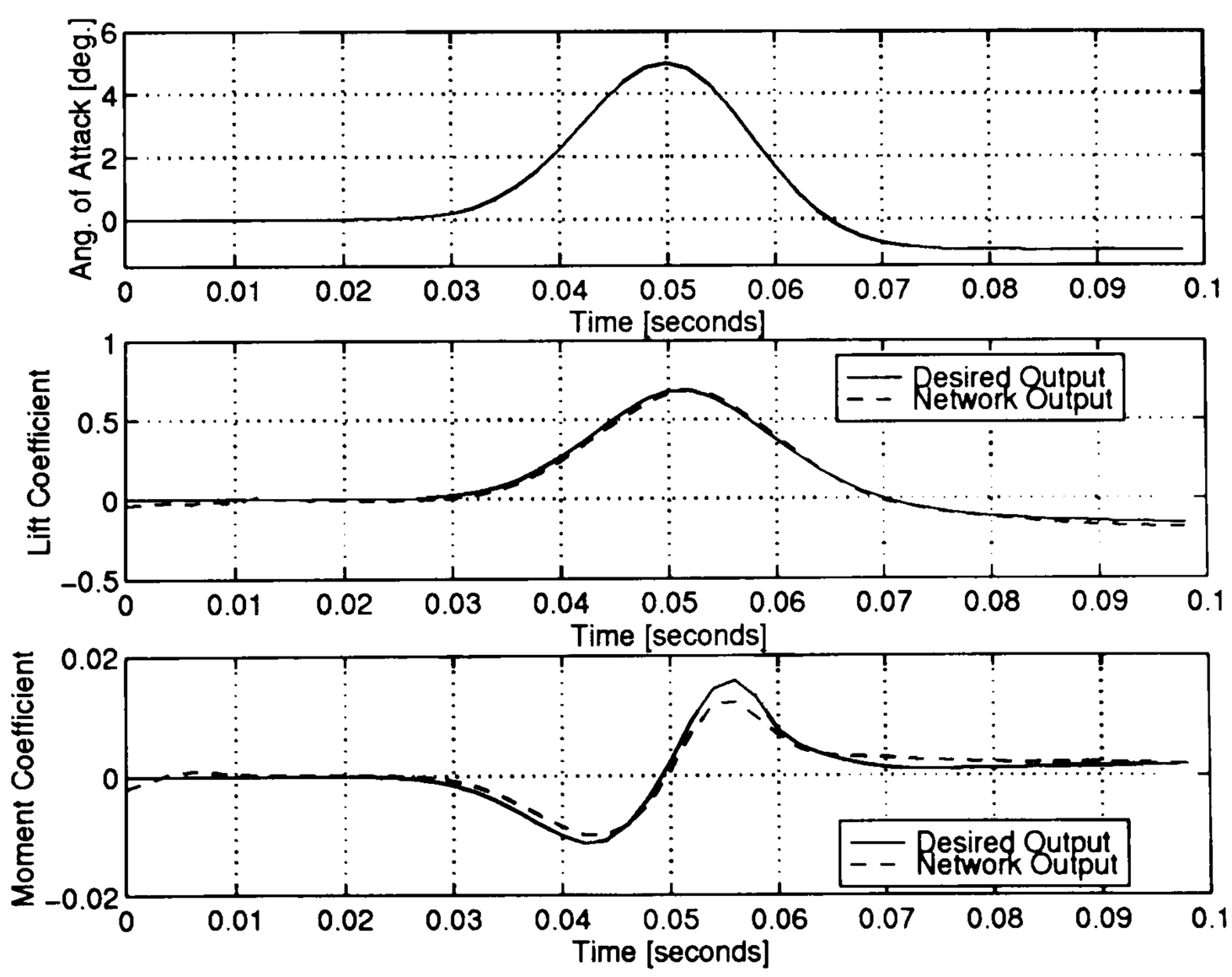


Figure 4.30: FIR network model of unsteady aerodynamic responses in the transonic regime (fixed $M = 0.65$) to arbitrary motion history beyond the training limits of incidence angle.

4.4.2 Unsteady Aerodynamic Response in Transonic Flow over a Range of Mach Numbers

A multi-layer functional representation of the unsteady aerodynamic response of a two-dimensional NACA 0012 airfoil operating over a range of Mach numbers in the transonic regime is identified. The representation considers the functional relationship between lift force coefficient, $C_L(t)$, and pitch moment coefficient at 25% chord length, $C_{m_{1/4}}(t)$, and the angle of attack histories, α_t, over a range of Mach numbers ($0.625 < M < 0.725$). Equation 4.5 describes the functional representation used.

$$\left\{ \begin{array}{c} C_L(t) \\ C_{m_{1/4}}(t) \end{array} \right\} = \mathcal{MF}[\alpha_t, M] \tag{4.5}$$

As in Section 4.4.1, three characteristic incidence motion histories are considered, but here each one is considered at a different Mach number. The training sets are presented in Table 4.11, and the basic problem features are the same as in Figure 4.17. In addition, the training sets are presented to the identification algorithm in terms of normalised values with respect to the maximum value of the aerodynamic responses. The non-linear behaviour of the unsteady aerodynamic response in this problem can be observed in Appendix B, where the existence and dynamic excursion of shock waves is apparent in each training case. To ensure adequate representation of the input motion histories and output aerodynamic responses the sampling rate is selected 0.002 s.

For the network model, the maximum complexity FIR neural network in the population is defined by 2 hidden layers and 10 neurons per hidden layer. A maximum time-delay per connection (τ_{max}) of 4 is assumed. The identification process is executed in 430,000 generations partitioned into three stages, in order to improve the convergence rate by changing (via a trial-and-error approach) certain training parameters from one stage to another. Table 4.12 presents the complete set of

training parameters used in the identification problem.

Characteristic motion	M	Range
sinusoidal	0.625	$\alpha_{mean} = 0°$; amplitude= 4.5°, frequency 10 Hz
	0.675	$\alpha_{mean} = 0°$; amplitude= 3.0°, frequency 10 Hz
	0.725	$\alpha_{mean} = 0°$; amplitude= 1.5°, frequency 10 Hz
sigmoidal ramp-up	0.625	$\alpha_{min} = 0°$; $\alpha_{max} = 4.5°$
	0.675	$\alpha_{min} = 0°$; $\alpha_{max} = 3.0°$
	0.725	$\alpha_{min} = 0°$; $\alpha_{max} = 1.5°$
pulse-down	0.625	$\alpha_{initial} = 0°$: $\alpha_{pulse} = -4.5°$; $\alpha_{final} = 0°$
	0.675	$\alpha_{initial} = 0°$; $\alpha_{pulse} = -3.0°$: $\alpha_{final} = 0°$
	0.725	$\alpha_{initial} = 0°$; $\alpha_{pulse} = -1.5°$: $\alpha_{final} = 0°$

Table 4.11: Training set motions for the identification of the unsteady aerodynamic response model for a range of Mach numbers in the transonic regime.

Training parameters	Value		
	Stage 1	Stage 2	Stage 3
population size	14	14	14
number of crossover points	7	13	5
P_t and P_n	0.5%	0.5%	0.5%
scaling factor for selection	2.0	2.0	2.0
perturbation constant, β	0.0001	0.0001	0.0001
number of cycles in updating weight/bias	5	5	5
number of steps before forced mutation	200	200	100
number of generations	200,000	50,000	180,000

Table 4.12: Training parameters for the identification of the unsteady aerodynamic response model for a range of Mach numbers in the transonic regime.

Figures 4.31 to 4.39 present a comparison between lift force coefficient and pitch moment coefficient responses obtained by the Euler CFD code and the respective FIR neural network outputs for each of the training sets after completion of the identification process.

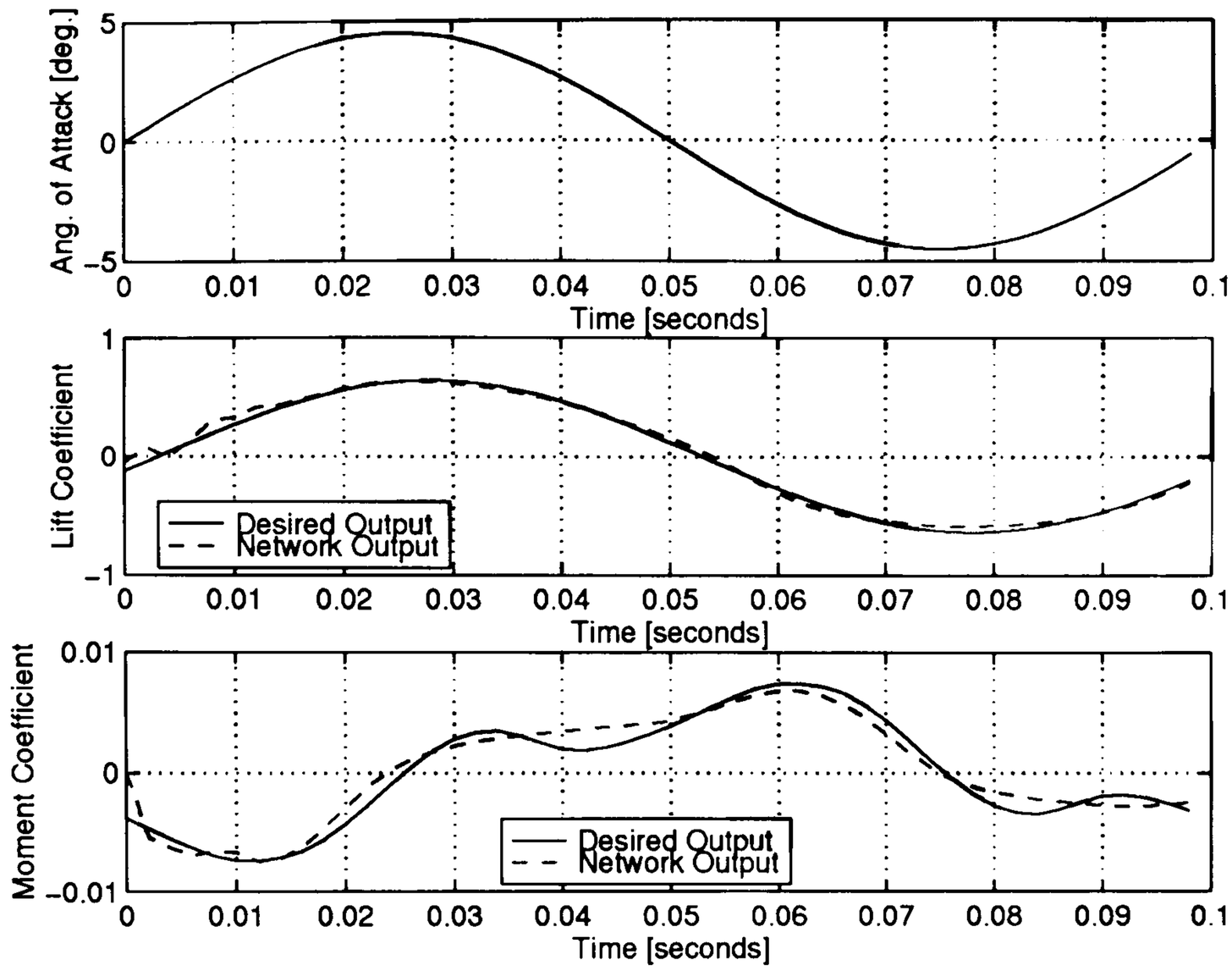


Figure 4.31: Identification of the unsteady aerodynamic response in the transonic regime ($M = 0.625$): Euler CFD code and FIR network outputs after training for the sinusoidal case.

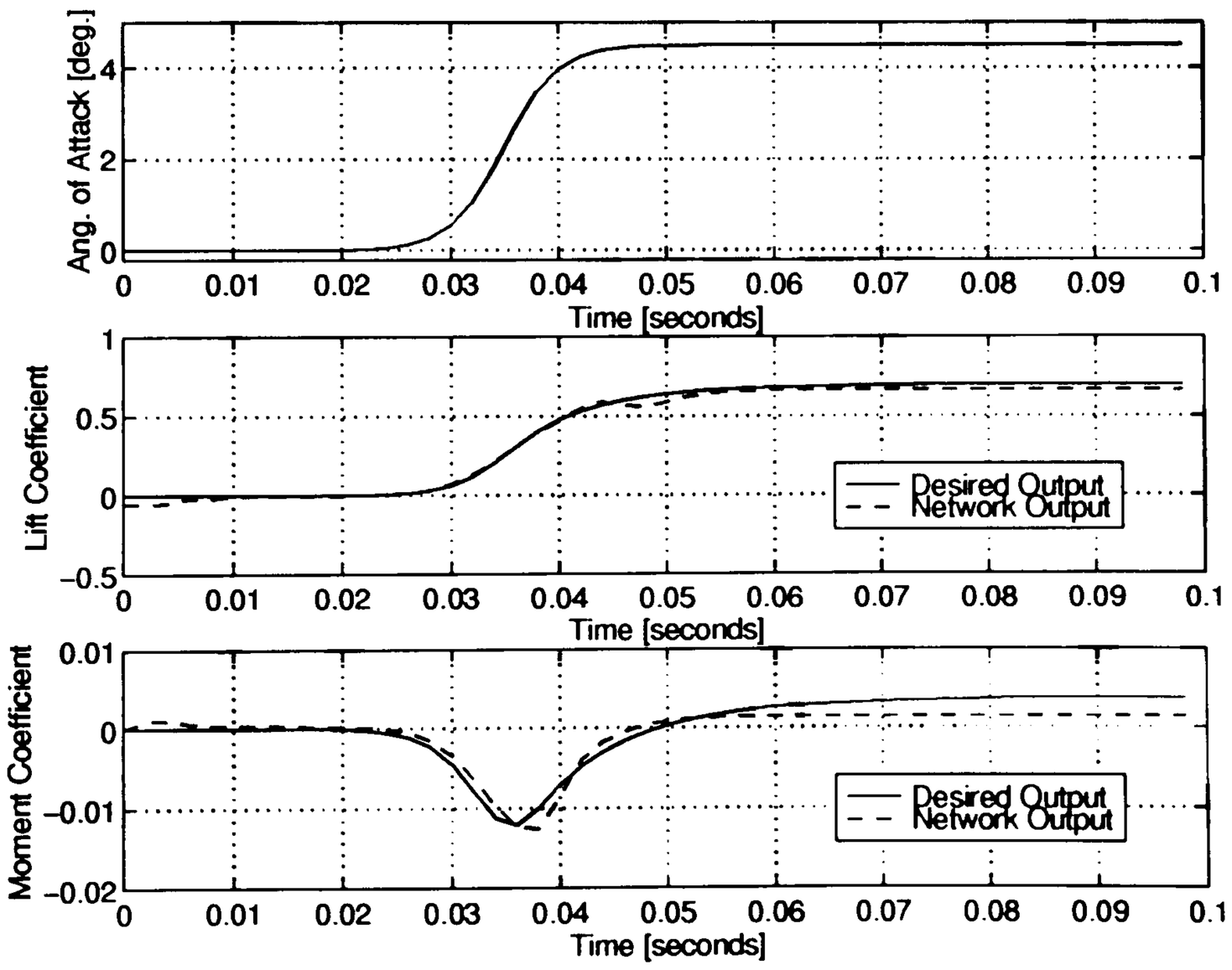


Figure 4.32: Identification of the unsteady aerodynamic response in the transonic regime ($M = 0.625$): Euler CFD code and FIR network outputs after training for the sigmoidal ramp-up case.

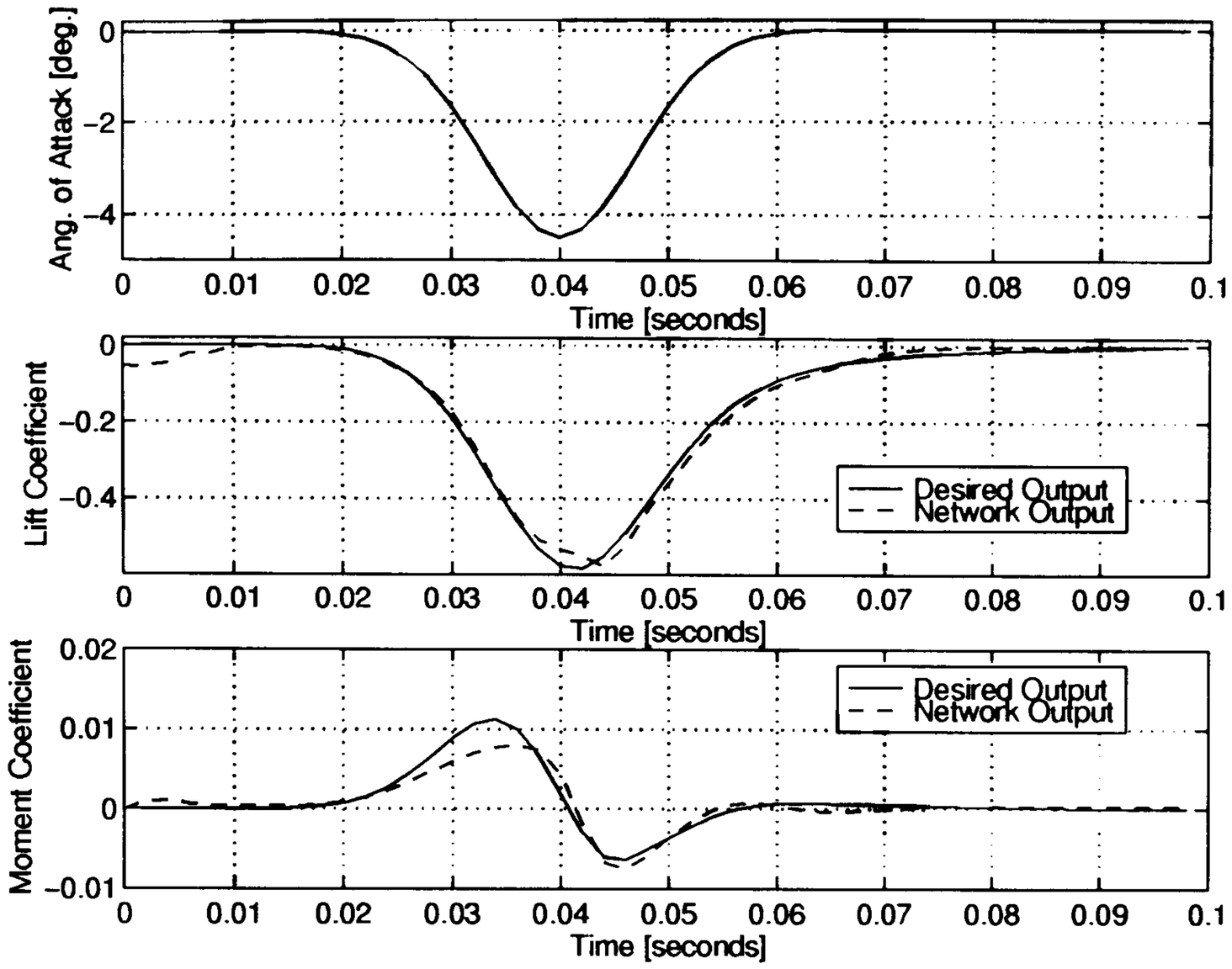


Figure 4.33: Identification of the unsteady aerodynamic response in the transonic regime ($M = 0.625$): Euler CFD code and FIR network outputs after training for the pulse-down case.

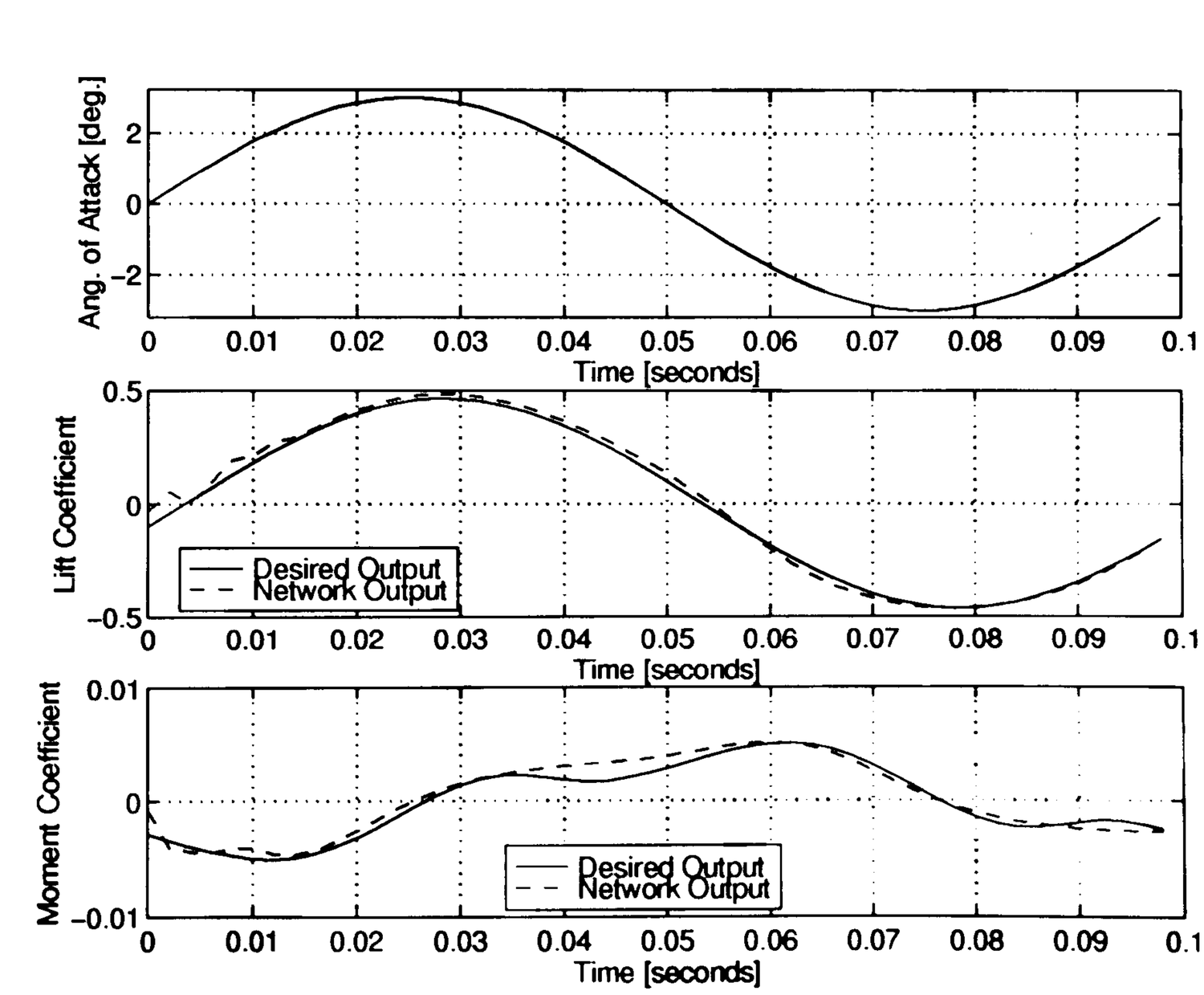


Figure 4.34: Identification of the unsteady aerodynamic response in the transonic regime ($M = 0.675$): Euler CFD code and FIR network outputs after training for the sinusoidal case.

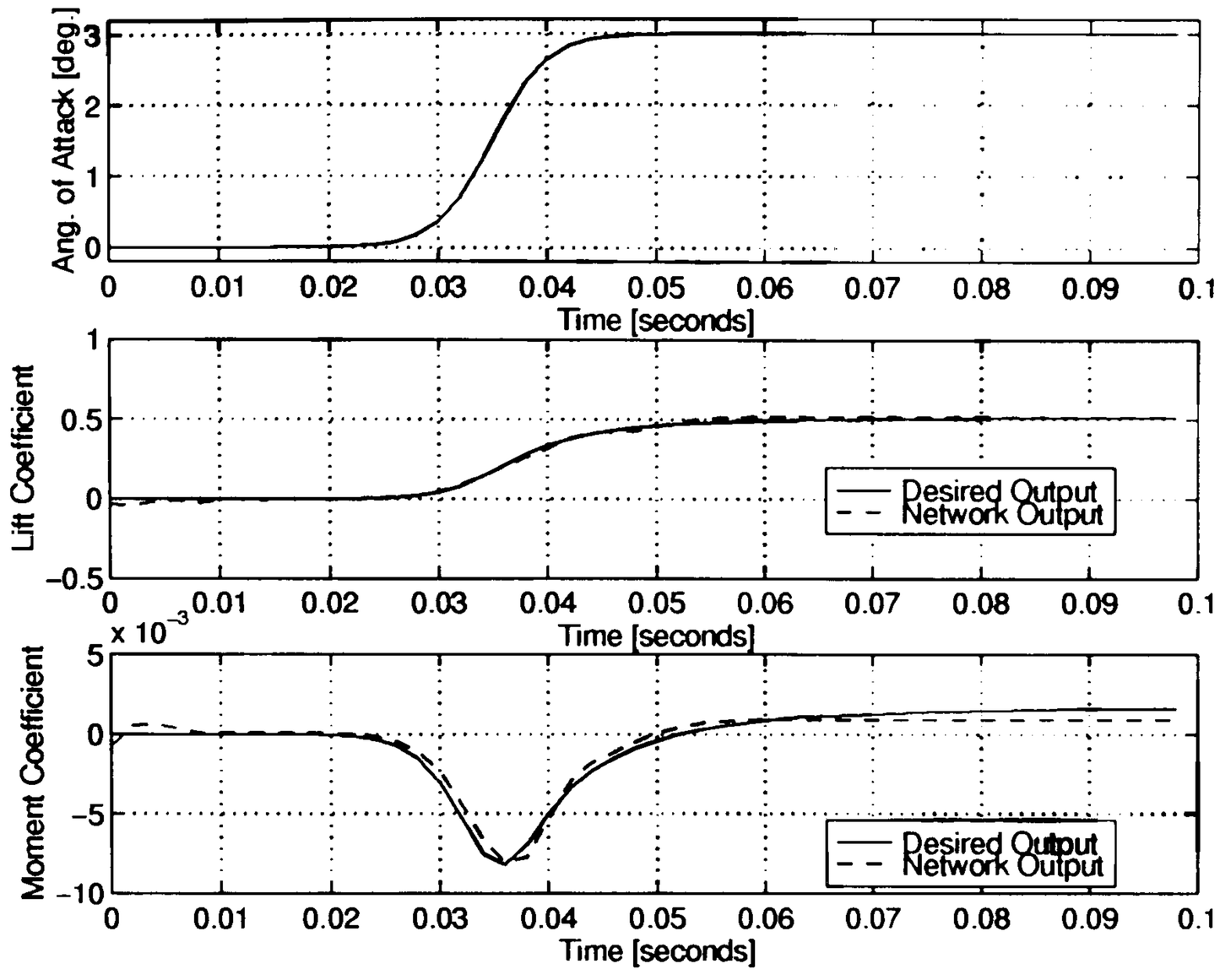


Figure 4.35: Identification of the unsteady aerodynamic response in the transonic regime ($M = 0.675$): Euler CFD code and FIR network outputs after training for the sigmoidal ramp-up case.

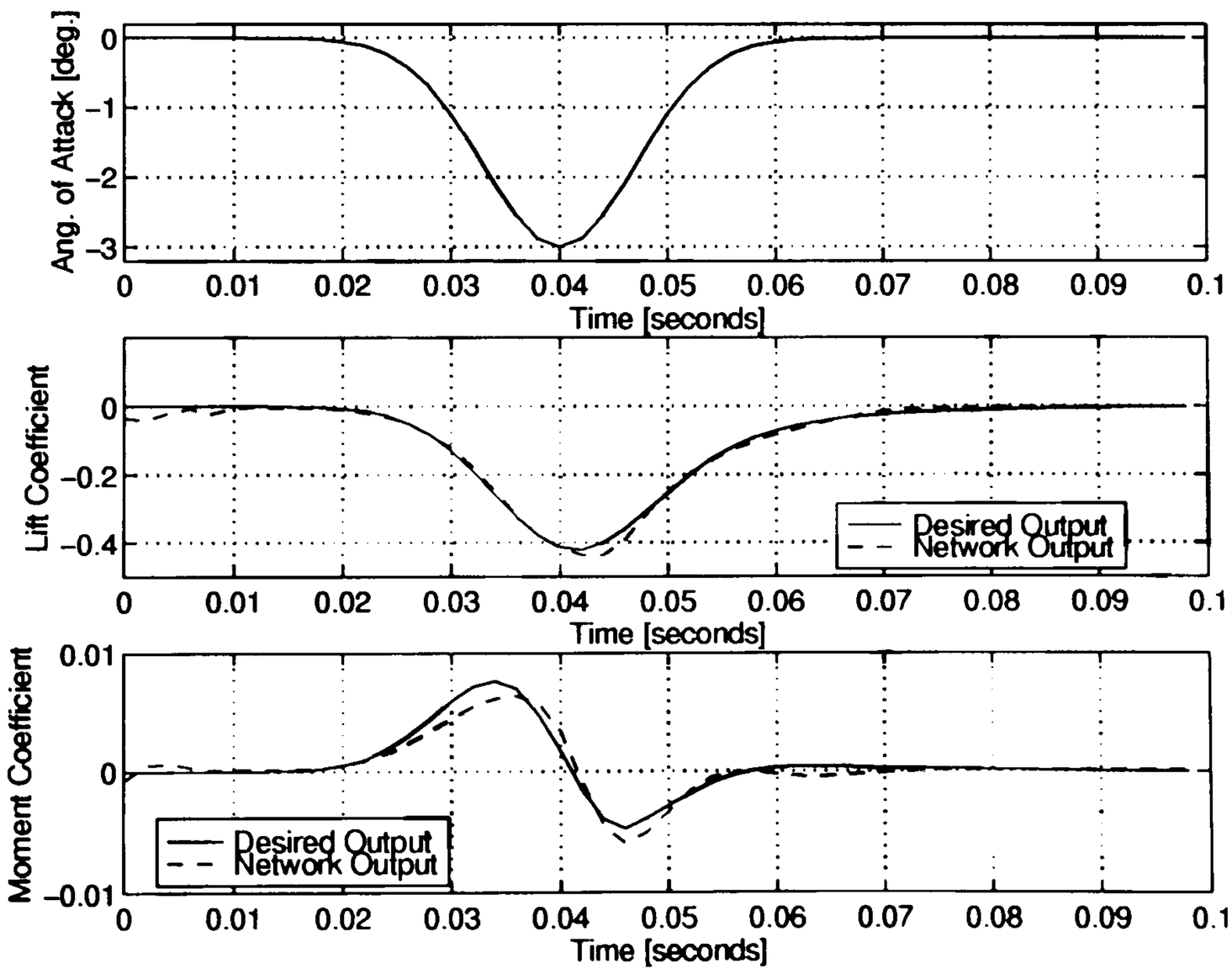


Figure 4.36: Identification of the unsteady aerodynamic response in the transonic regime ($M = 0.675$): Euler CFD code and FIR network outputs after training for the pulse-down case.

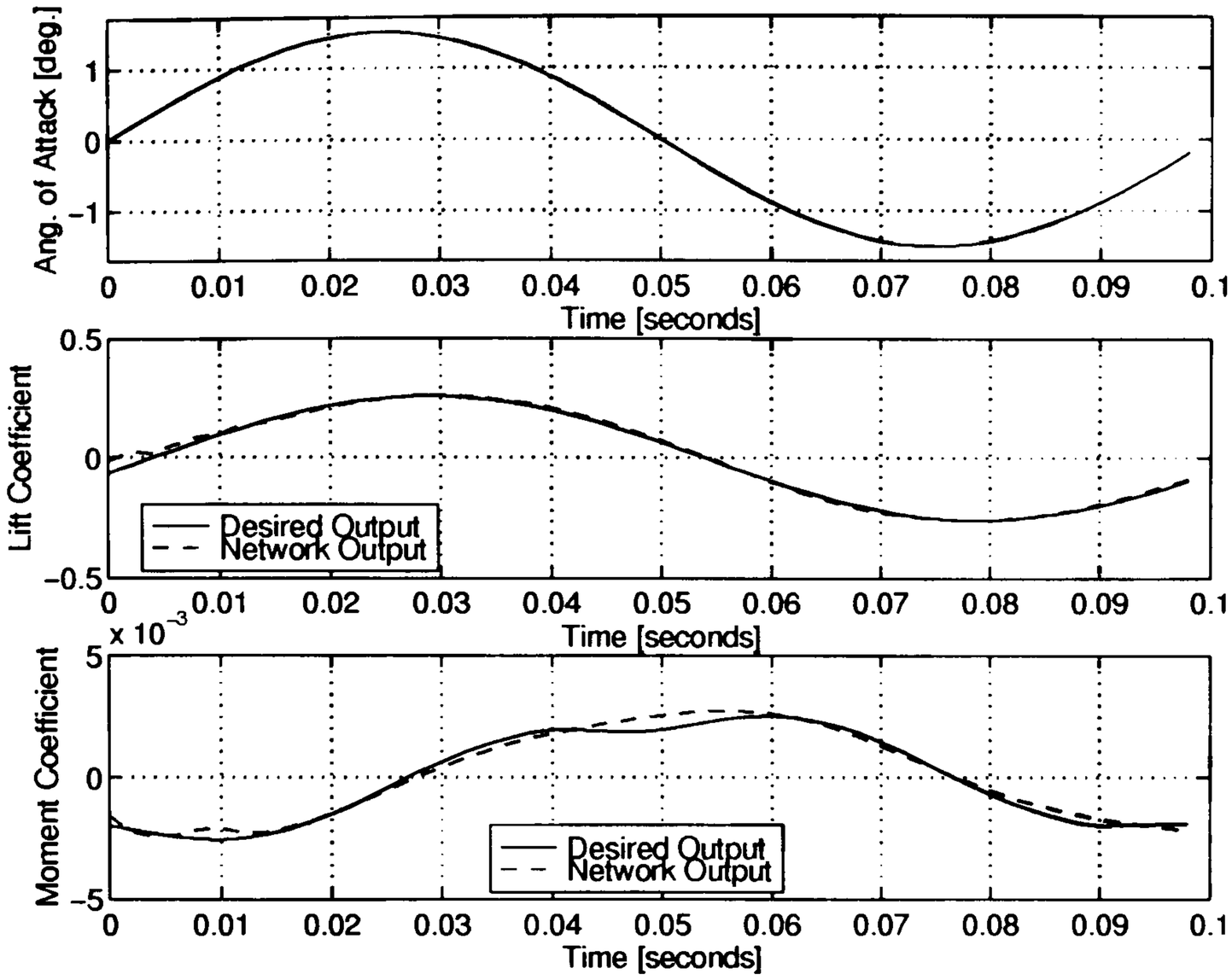


Figure 4.37: Identification of the unsteady aerodynamic response in the transonic regime ($M = 0.725$): Euler CFD code and FIR network outputs after training for the sinusoidal case.

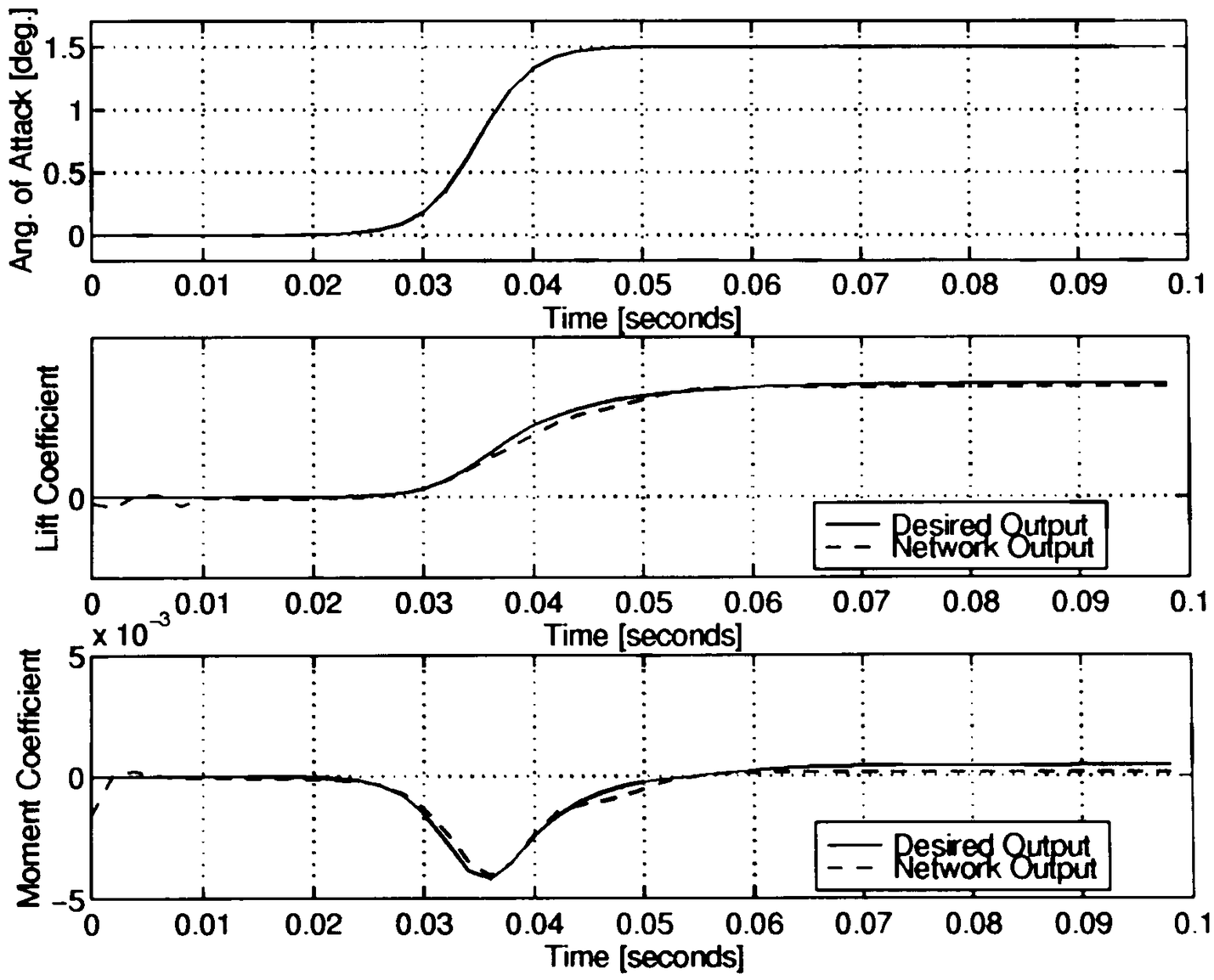


Figure 4.38: Identification of the unsteady aerodynamic response in the transonic regime ($M = 0.725$): Euler CFD code and FIR network outputs after training for the sigmoidal ramp-up case.

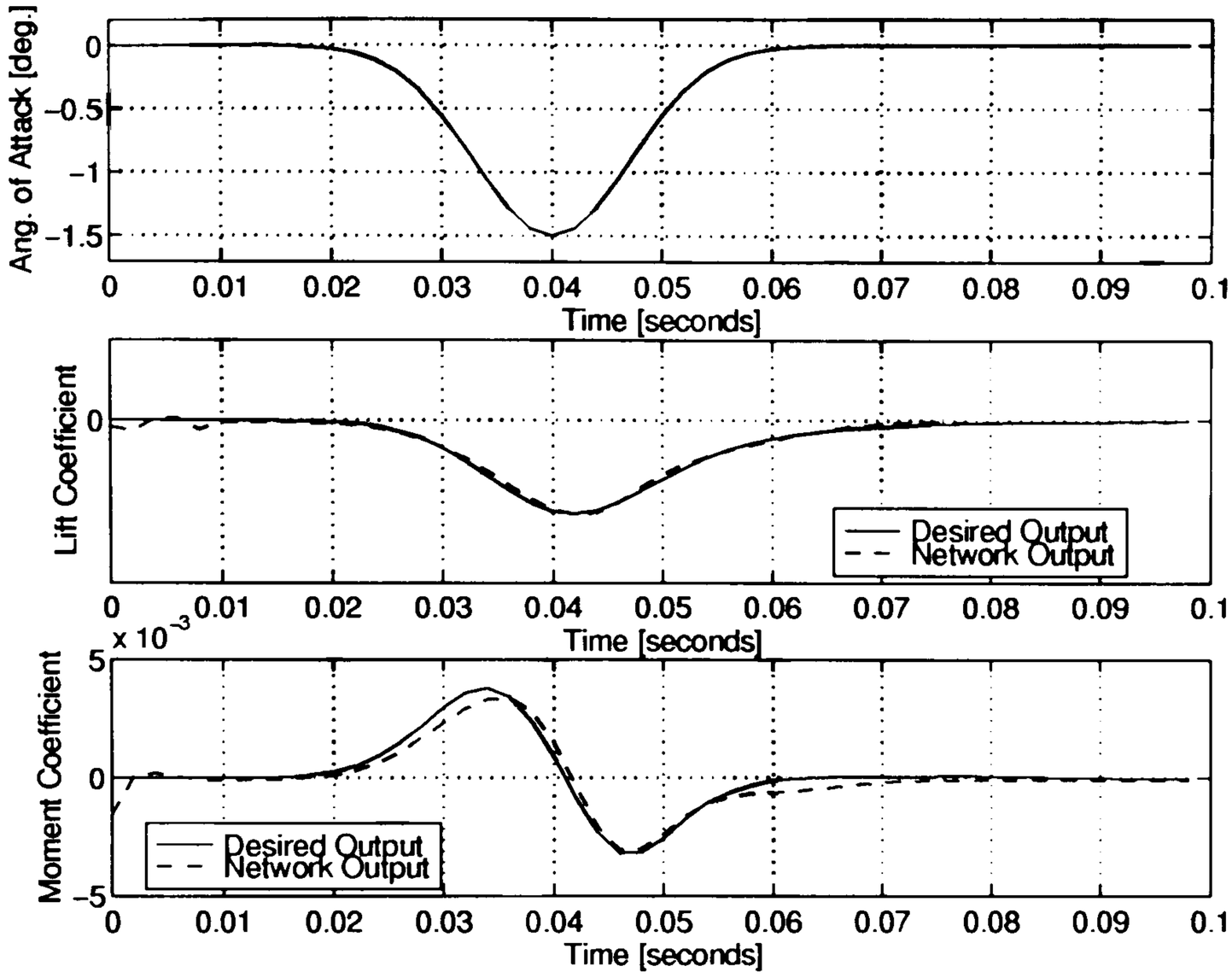


Figure 4.39: Identification of the unsteady aerodynamic response in the transonic regime ($M = 0.725$): Euler CFD code and FIR network outputs after training for the pulse-down case.

The architecture and time-delay distribution of the identified FIR network model are presented in Figure 4.40, while the respective values of weight and bias are presented in Tables 4.13, 4.14, and 4.15.

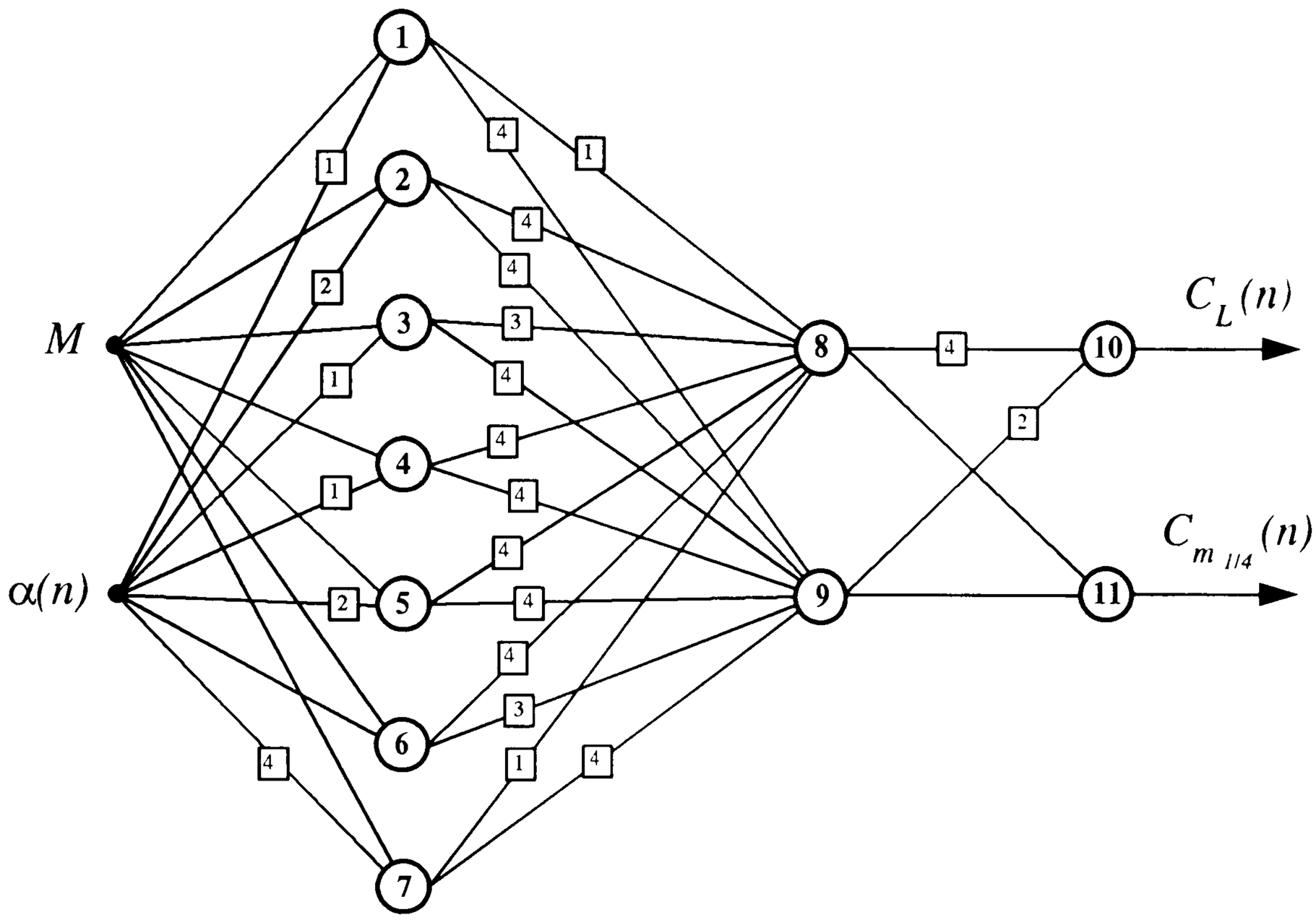


Figure 4.40: Identified FIR neural network model of the unsteady aerodynamic response for a range of Mach numbers in the transonic regime ($0.625 < M < 0.725$).

		1	2	3	4	5	6	7	8	9	10	11
M		0.724	-0.882	-0.415	-0.431	0.203	-0.364	-0.500				
α		-0.055	0.154	0.003	-1.130	0.543	0.014	-0.763				
	(1)	0.220	0.707	-0.210	1.247	0.121	-	0.784				
	(2)	-	-0.399	-	-	-0.164	-	-0.742				
	(3)	-	-	-	-	-	-	0.376				
	(4)	-	-	-	-	-	-	0.643				
1									-0.521	0.422		
	(1)								0.128	-0.791		
	(2)								-	0.163		
	(3)								-	0.090		
	(4)								-	0.076		
2									0.202	-0.762		
	(1)								0.039	0.755		
	(2)								0.068	-0.142		
	(3)								-0.896	0.251		
	(4)								0.121	0.120		
3									0.658	-0.401		
	(1)								0.125	0.169		
	(2)								-0.038	-0.176		
	(3)								0.032	-0.033		
	(4)								-	-0.039		
4									0.509	-1.007		
	(1)								-0.465	-0.587		
	(2)								0.792	0.462		
	(3)								-0.111	0.179		
	(4)								-0.285	0.311		
5									-0.197	-0.028		
	(1)								0.111	-0.233		
	(2)								0.193	0.280		
	(3)								0.221	-0.005		
	(4)								0.143	-0.052		
6									-0.607	0.981		
	(1)								0.014	-0.850		
	(2)								-0.738	0.634		
	(3)								0.859	-0.316		
	(4)								0.014	-		
7									-0.334	-0.054		
	(1)								0.194	-0.411		
	(2)								-	-0.122		
	(3)								-	-0.081		
	(4)								-	-0.033		

Table 4.13: Weight values in the identified FIR network model of the unsteady aerodynamic response for a range of Mach numbers of the transonic regime.

		1	2	3	4	5	6	7	8	9	10	11
8											-0.550	0.004
	(1)										-0.495	-
	(2)										0.013	-
	(3)										0.009	-
	(4)										-0.020	-
9											0.076	-0.013
	(1)										-0.318	-
	(2)										0.110	-
	(3)										-	-
	(4)										-	-

Table 4.14: Weight values in the identified FIR network model of the unsteady aerodynamic response for a range of Mach numbers of the transonic regime: cont'd.

	1	2	3	4	5	6	7	8	9	10	11
θ	-0.011	0.404	0.261	0.198	-0.401	0.137	-0.062	0.391	-0.825	0.162	-0.008

Table 4.15: Bias values of each neuron of the identified FIR network model of the unsteady aerodynamic response for a range of Mach numbers in the transonic regime.

Figure 4.41 presents the convergence characteristics, in steps of ten generations, of the training process in terms of the normalised sum of squared errors between the Euler CFD code and adapted FIR neural network outputs of the best individual in the population with respect to the initial error value, which is determined by the inverse fitness value of the best individual in the population. The generalisation properties of the identified FIR network model are examined in Figures 4.42 to 4.51, by presenting the network model with arbitrary incidence motion histories and Mach numbers and comparing the output of the FIR network model with correspondent aerodynamic responses generated by the Euler CFD code.

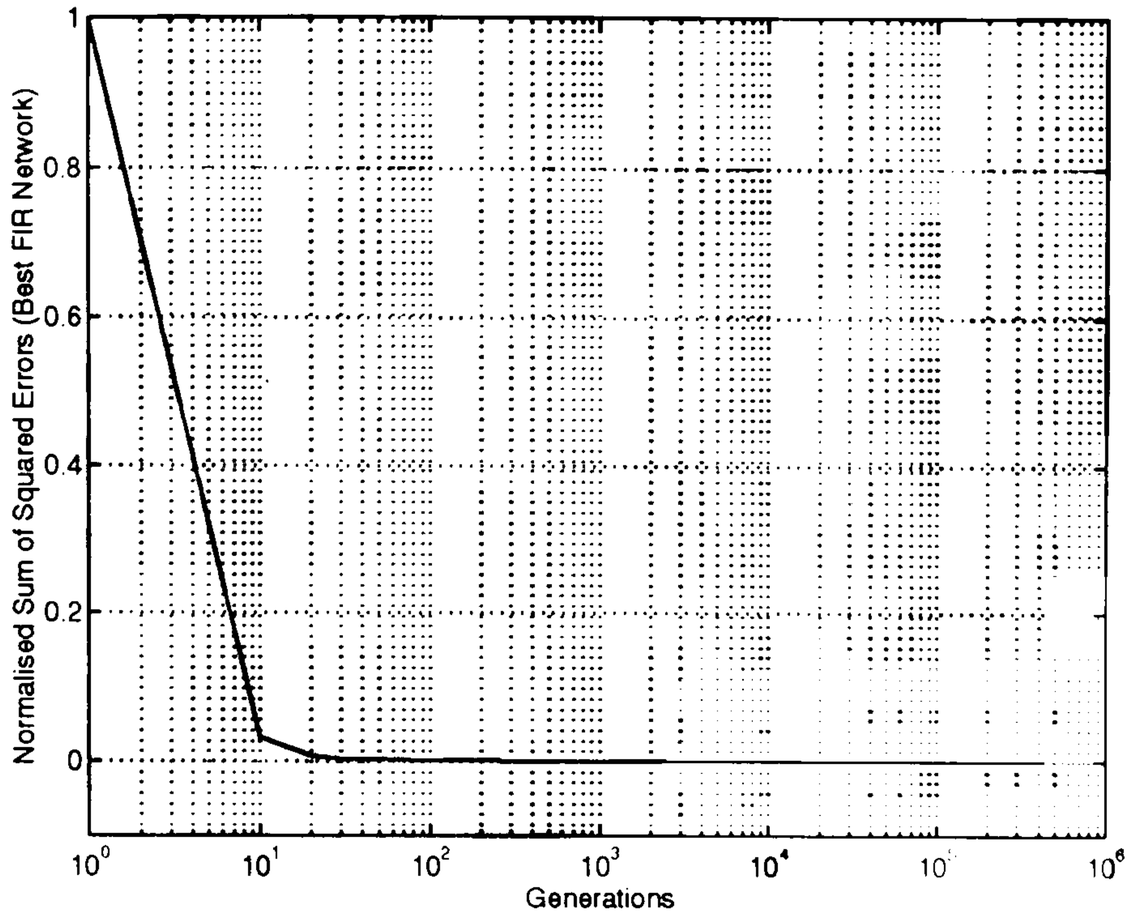


Figure 4.41: Error decay during the identification of the FIR network model of the unsteady aerodynamic response for a range of Mach numbers in the transonic regime.

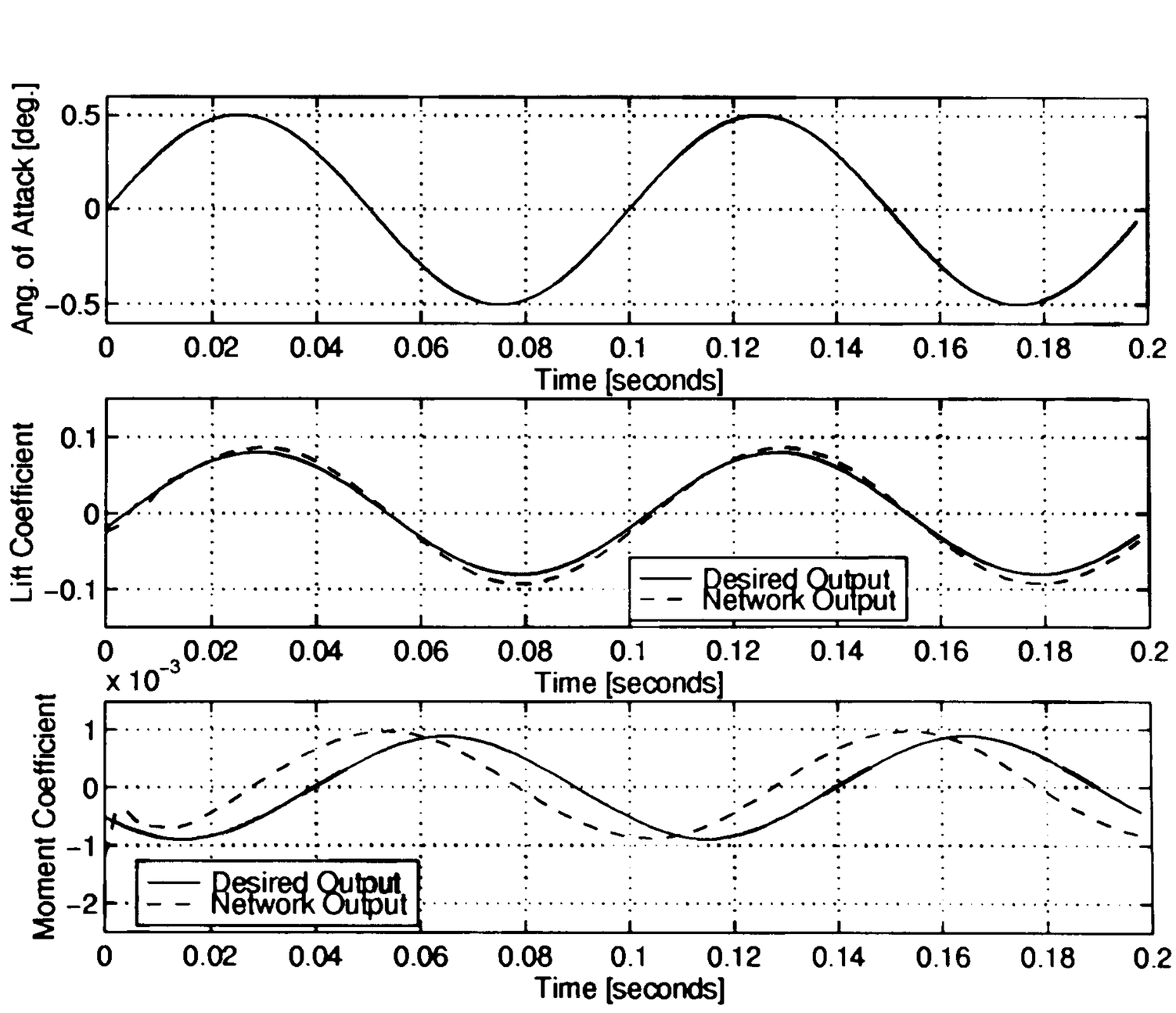


Figure 4.42: FIR network model of the unsteady aerodynamic response in the transonic regime to arbitrary motion history in the linear range ($M = 0.7$).

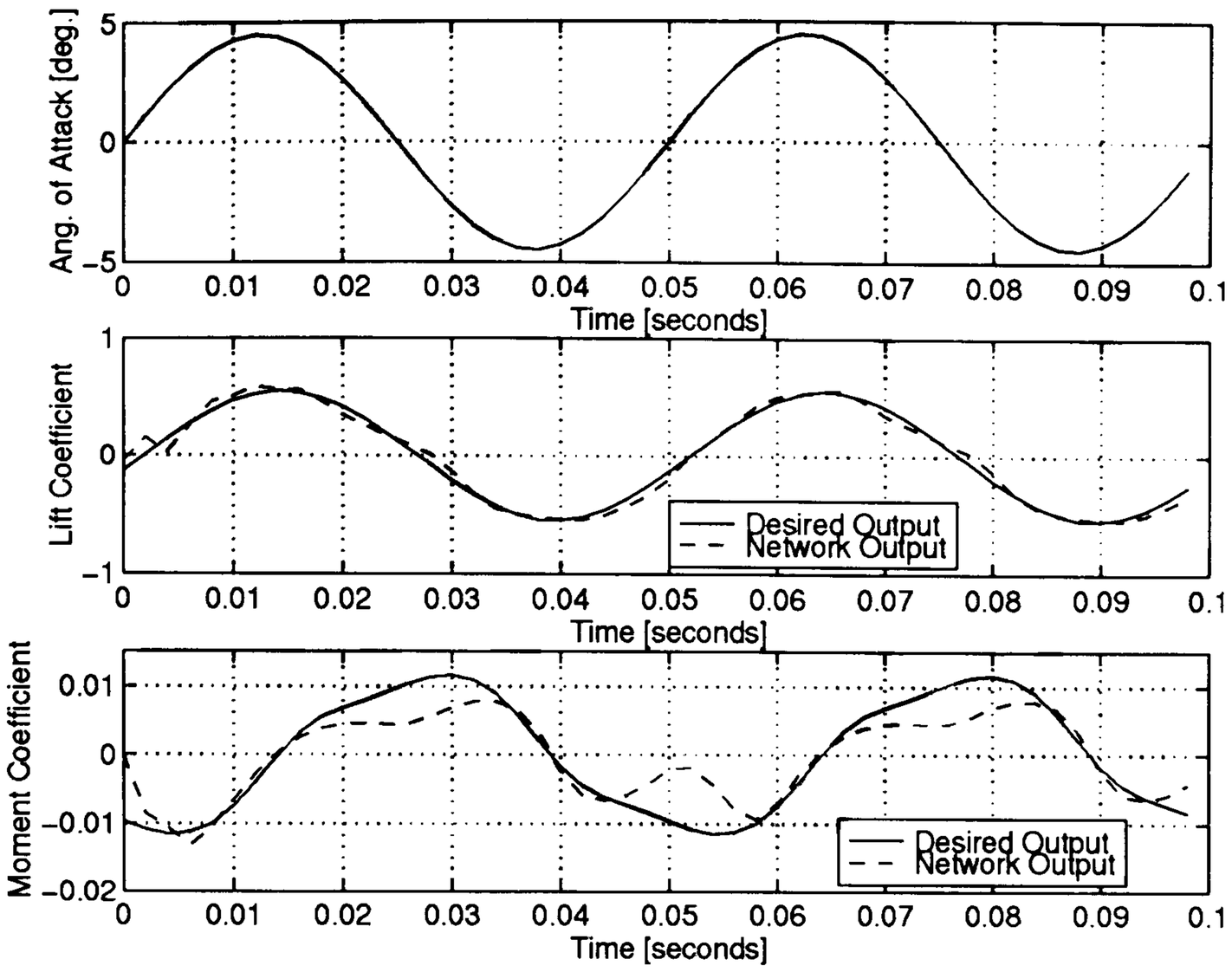


Figure 4.43: FIR network model of the unsteady aerodynamic response in the transonic regime to arbitrary motion history at higher frequency than that of the training sets ($M = 0.625$).

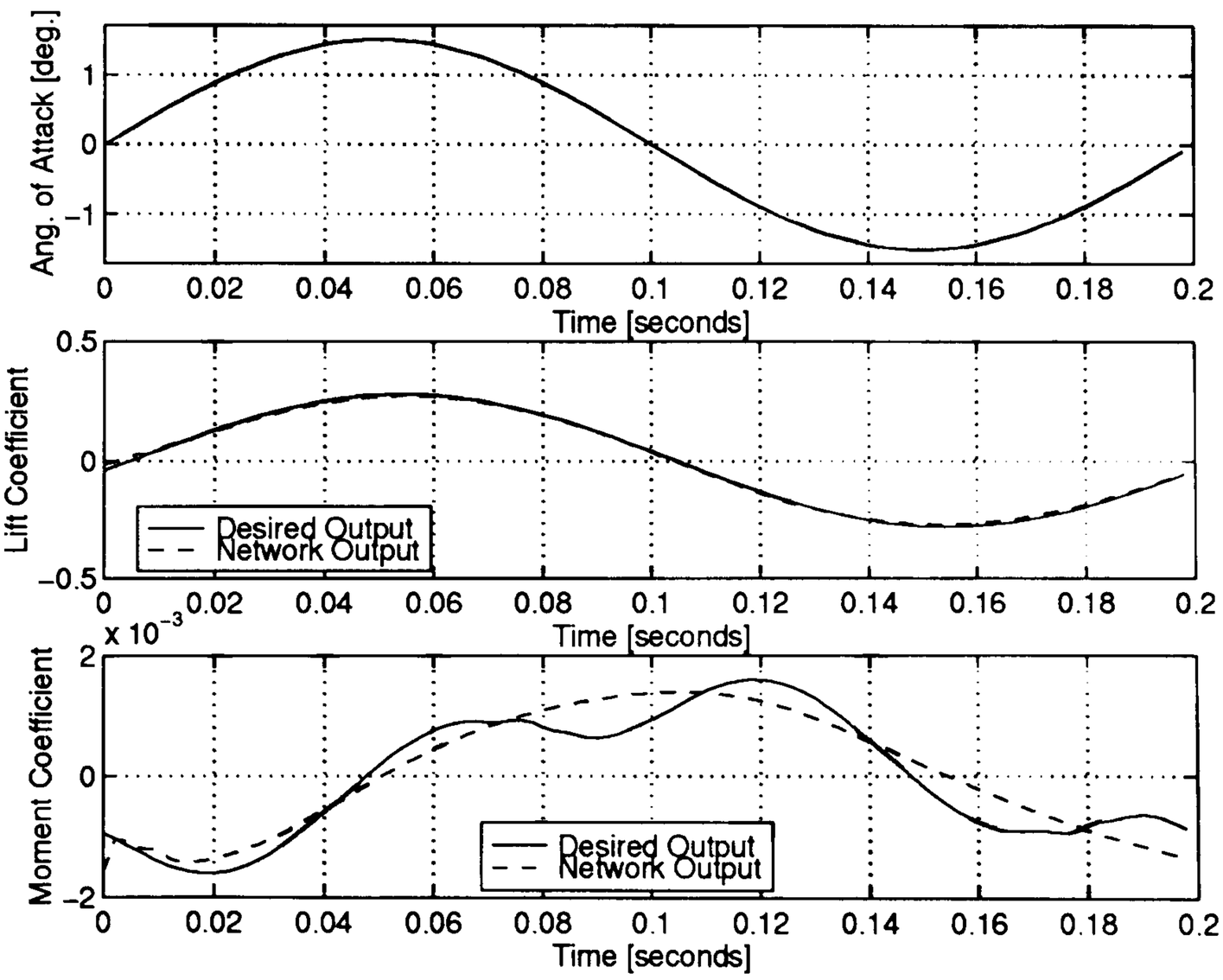


Figure 4.44: FIR network model of the unsteady aerodynamic response in the transonic regime to arbitrary motion history at lower frequency than that of the training sets ($M = 0.725$).

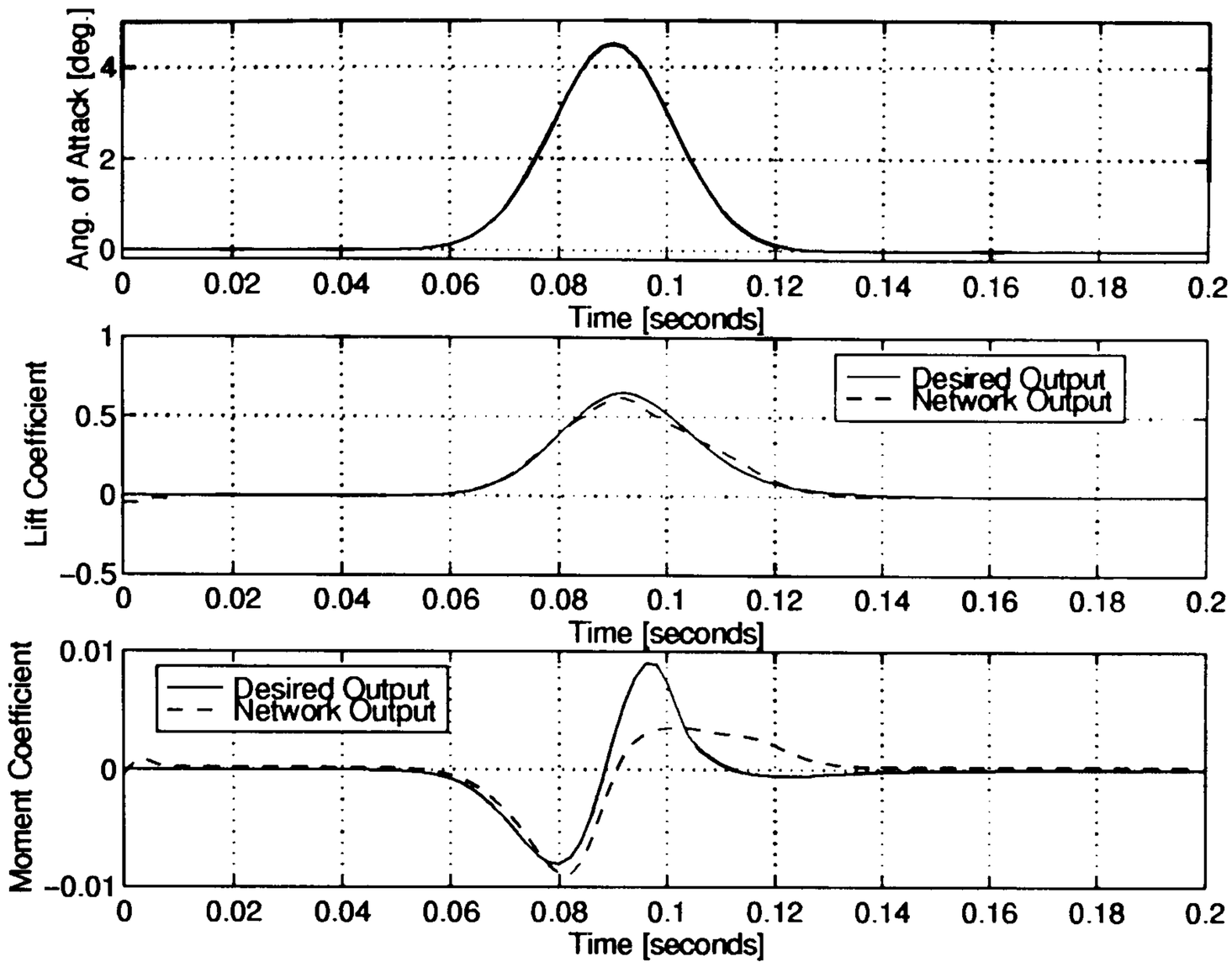


Figure 4.45: FIR network model of the unsteady aerodynamic response in the transonic regime to arbitrary motion history and Mach number ($M = 0.65$).

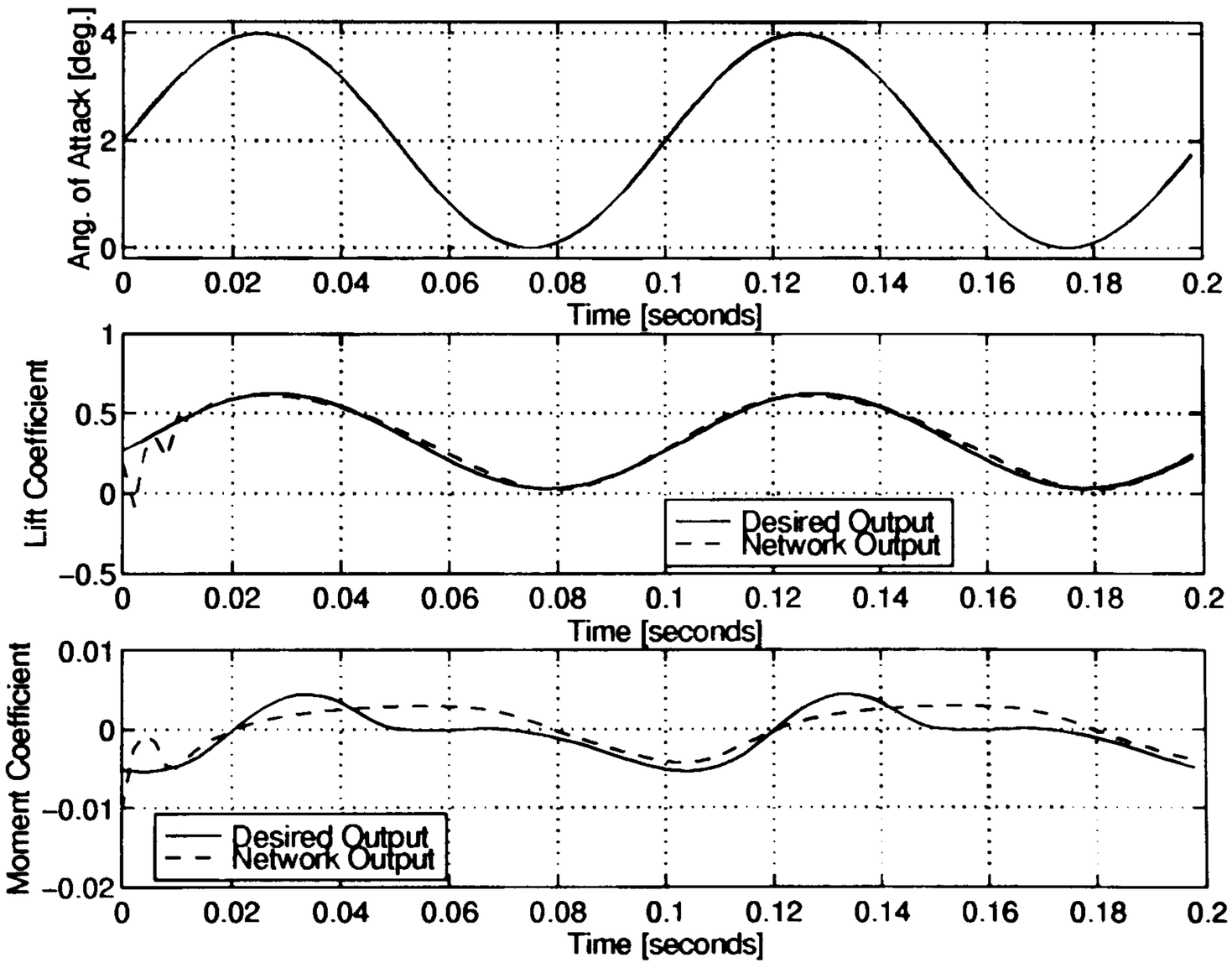


Figure 4.46: FIR network model of the unsteady aerodynamic response in the transonic regime to arbitrary motion history and Mach number ($M = 0.65$).

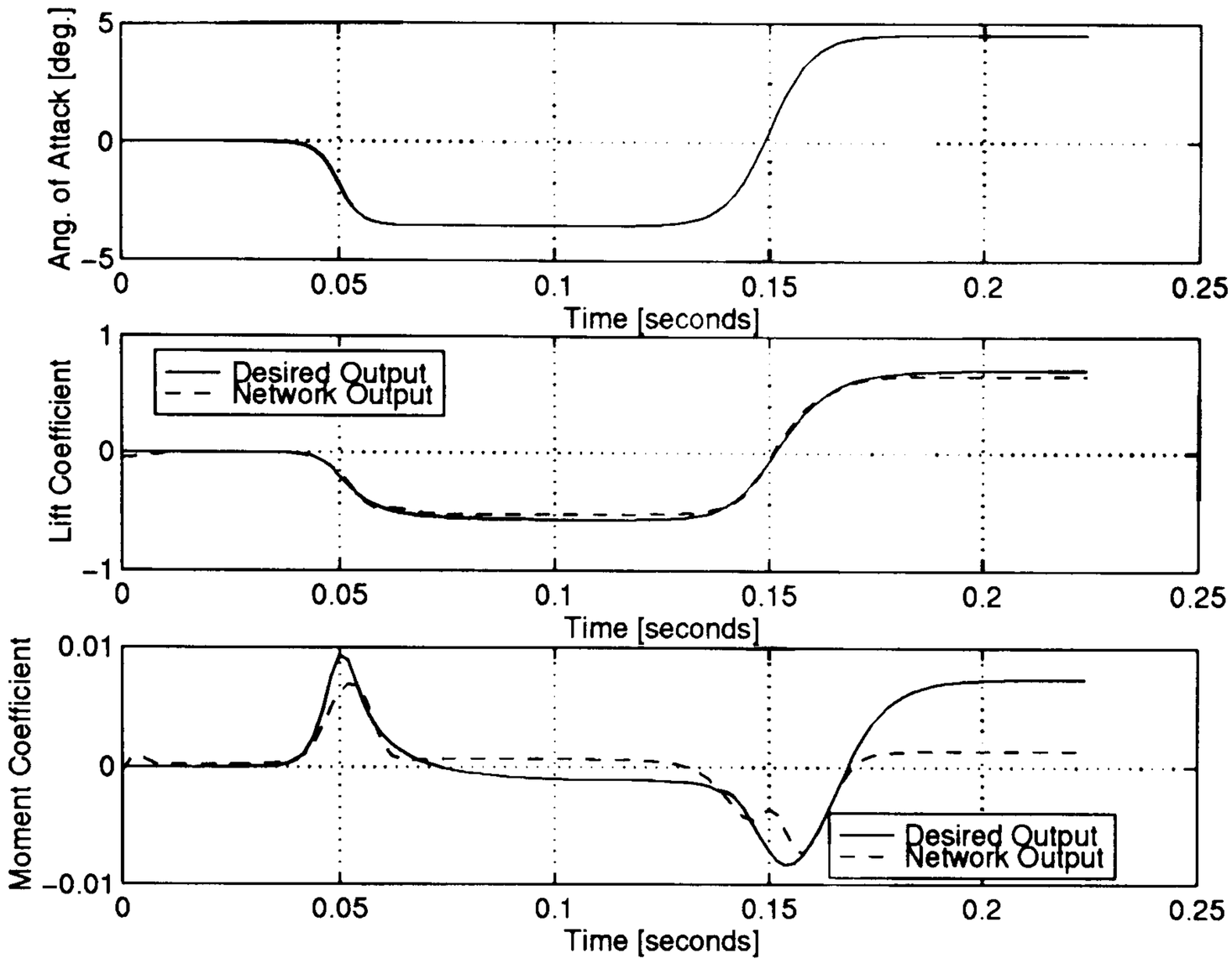


Figure 4.47: FIR network model of the unsteady aerodynamic response in the transonic regime to arbitrary motion history and Mach number ($M = 0.65$).

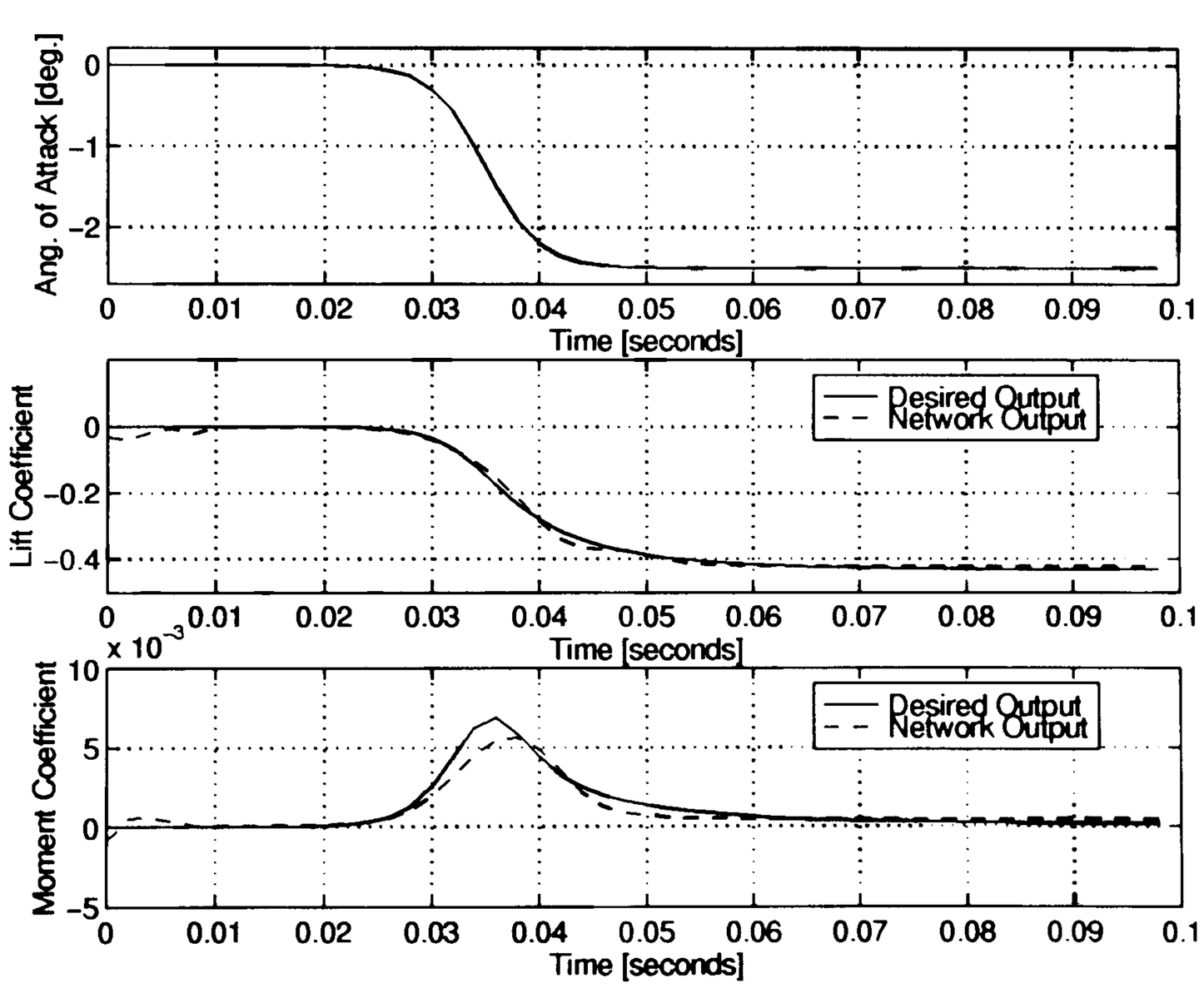


Figure 4.48: FIR network model of the unsteady aerodynamic response in the transonic regime to arbitrary motion history and Mach number ($M = 0.68$).

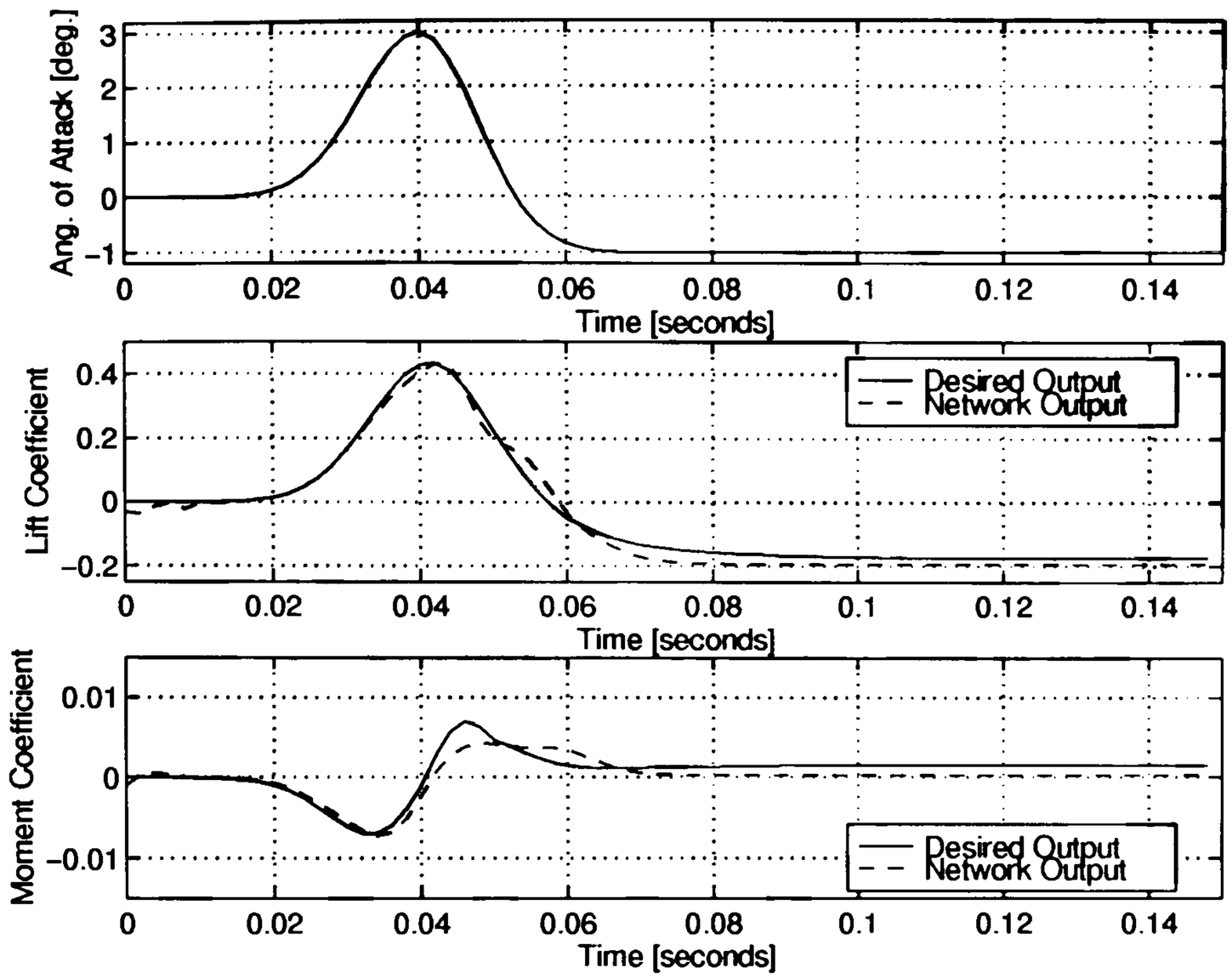


Figure 4.49: FIR network model of the unsteady aerodynamic response in the transonic regime to arbitrary motion history and Mach number ($M = 0.685$).

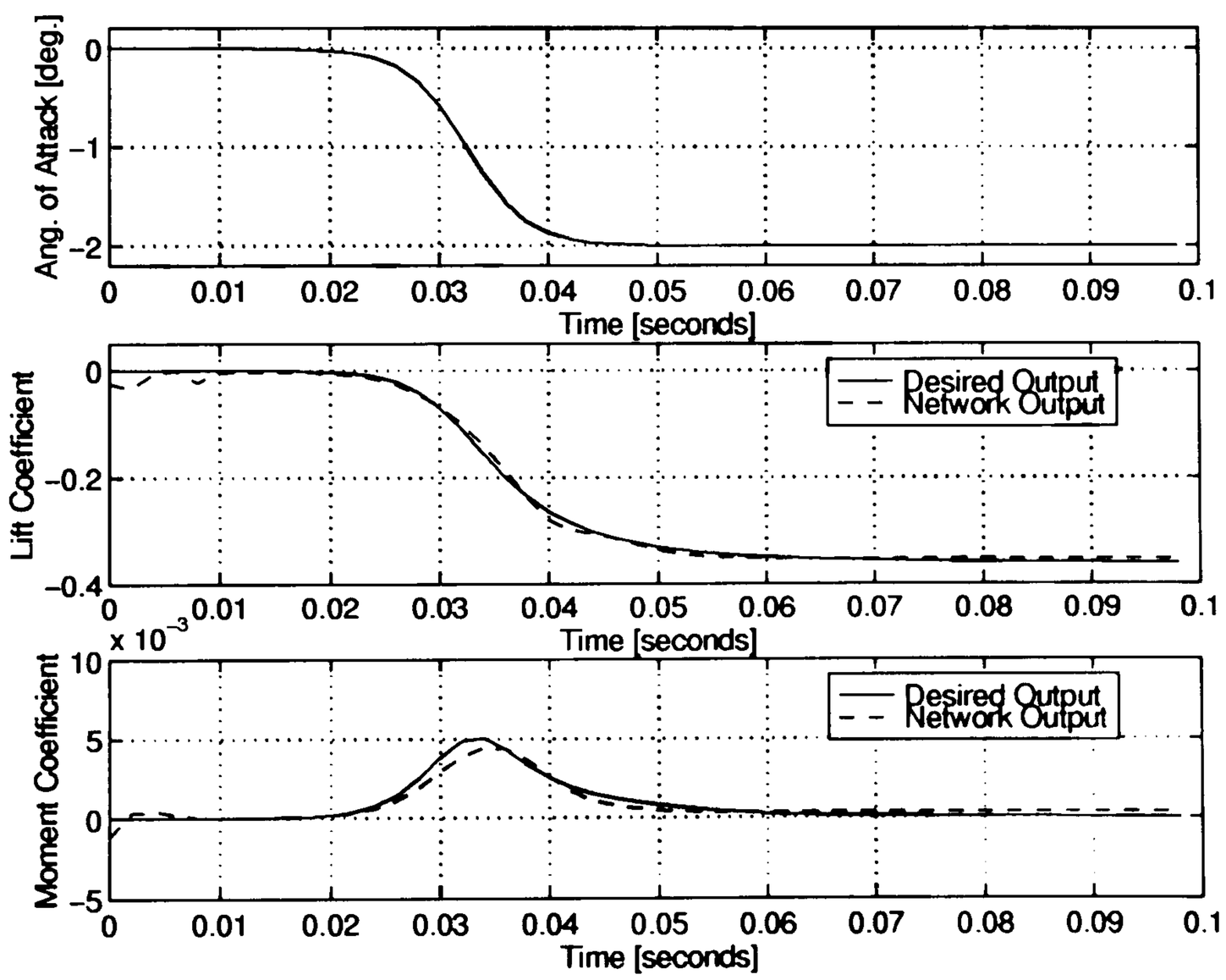


Figure 4.50: FIR network model of the unsteady aerodynamic response in the transonic regime to arbitrary motion history and Mach number ($M = 0.7$).

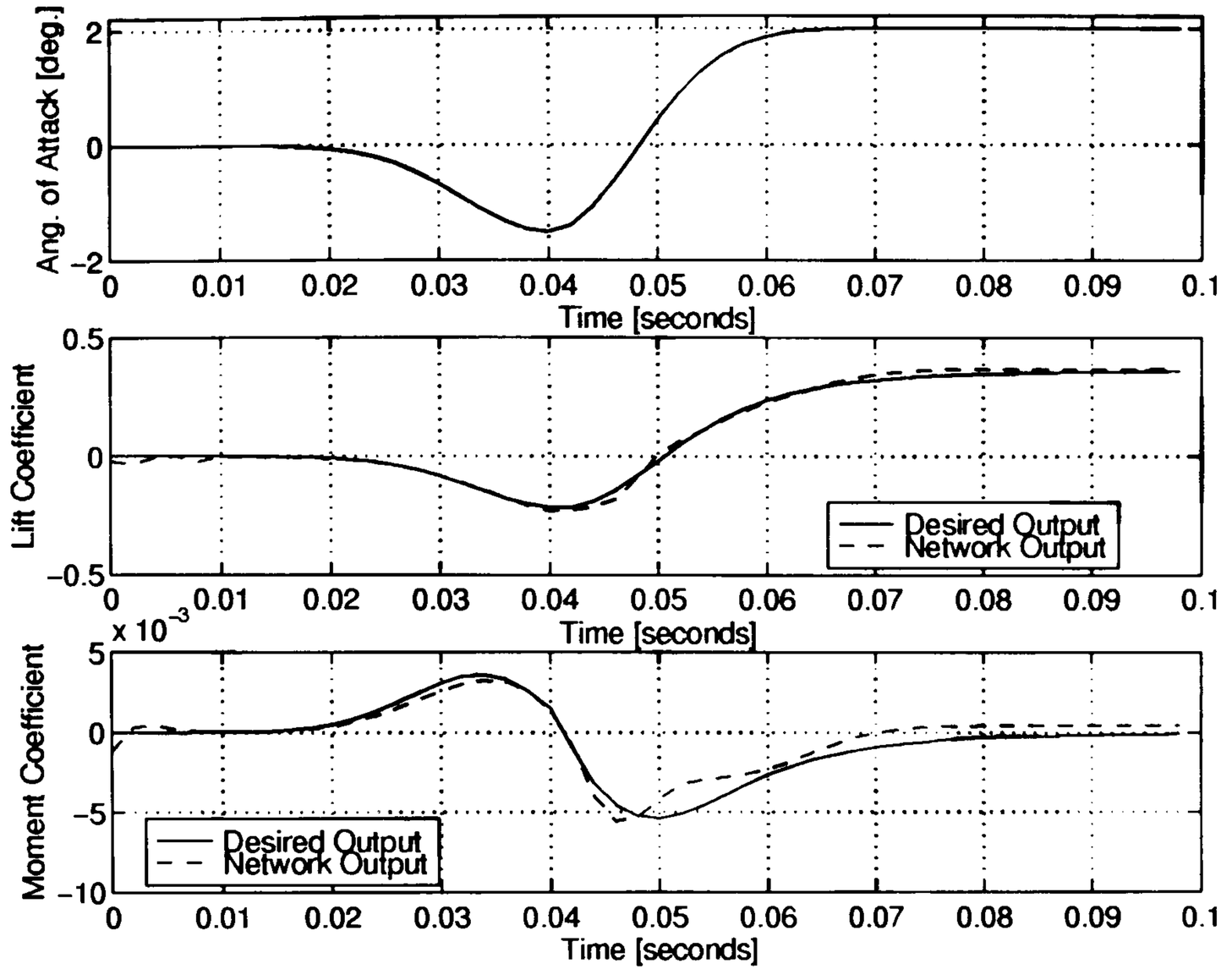


Figure 4.51: FIR network model of the unsteady aerodynamic response in the transonic regime to arbitrary motion history and Mach number ($M = 0.7$).

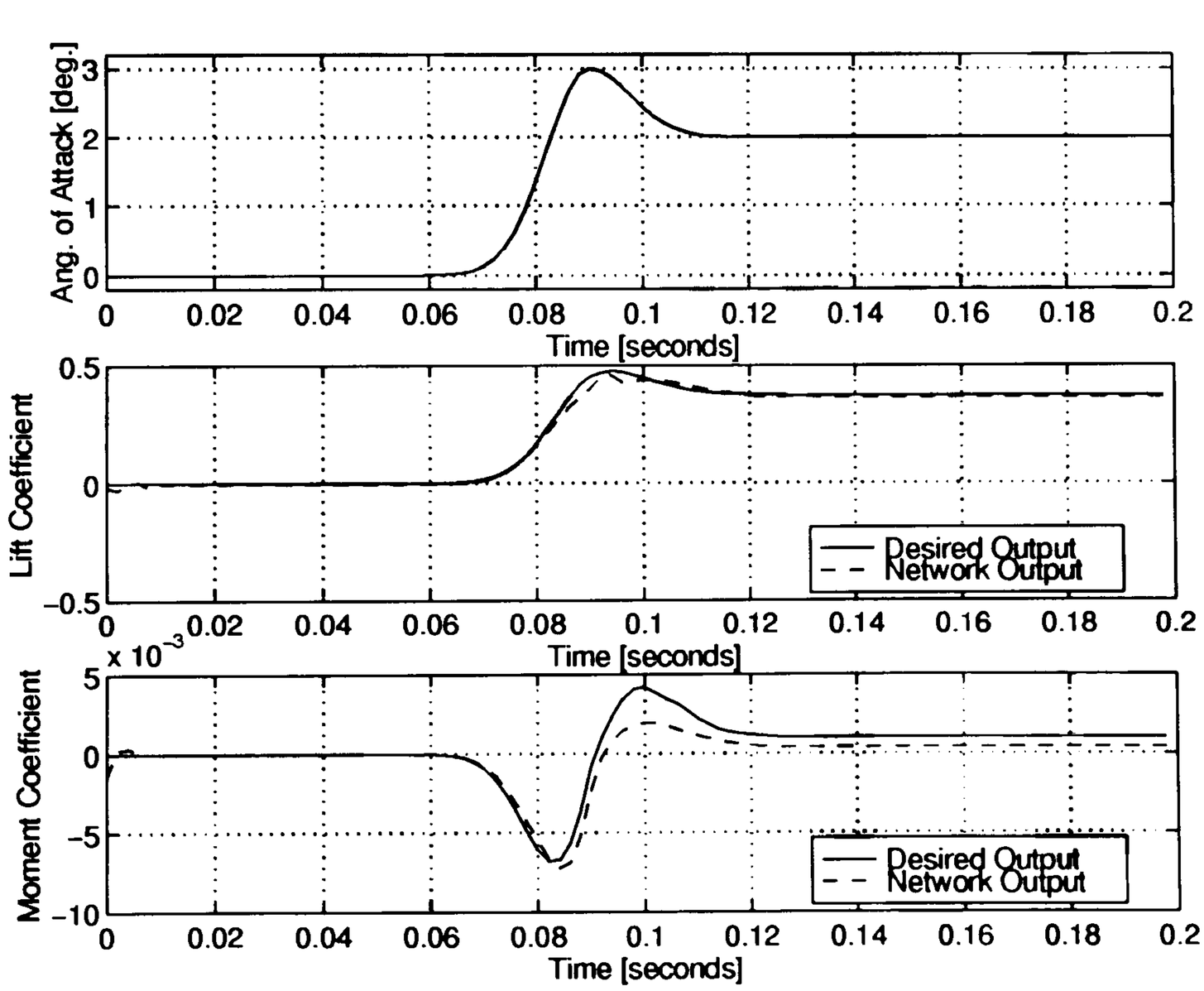


Figure 4.52: FIR network model of the unsteady aerodynamic response in the transonic regime to arbitrary motion history and Mach number ($M = 0.715$).

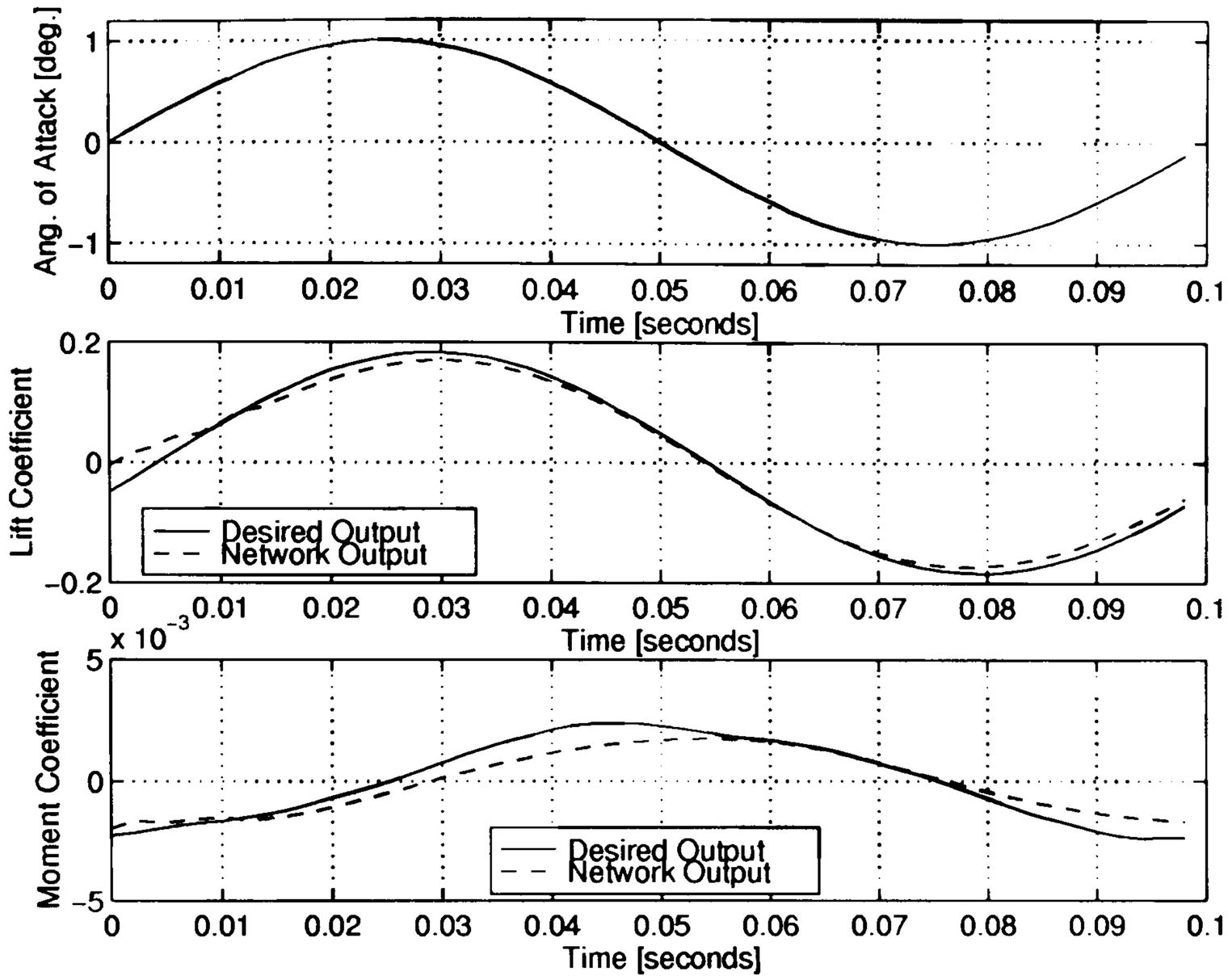


Figure 4.53: FIR network model of the unsteady aerodynamic response in the transonic regime to arbitrary motion history and Mach number beyond the training limits ($M = 0.75$).

4.5 Discussion

Both unsteady aerodynamic response models have shown encouraging results to endorse the use of multi-layer functional representations. The ability of the FIR neural network to capture the essential characteristics of both linear and non-linear unsteady aerodynamic behaviour can be observed in simulations for a range of motion histories and Mach numbers different from the ones used for training. In contrast to the considerable amount of time demanded by the training process to identify the model, the final network evaluations are fast enough to allow real-time predictions, justifying further applications in aeroelastic analysis and control design. Another

advantage of FIR neural networks is in the fact that the network parameters (that is, connectivity information and weight and bias values) demand minimal storage requirements.

The presence of two hidden layers in the final FIR neural network ensures functional complexity, as observed from all identified models. The typical time-delay distribution within the networks also provides features to the identified models that are consistent with the physical behaviour of unsteady flowfields. Among all the identified network architectures, a tendency in accummulating larger time-delay values between the two hidden layers has been observed. No connection has been found to relate time-delay distribution and values to the functional description of the unsteady aerodynamic response.

An adequate representation of the weakly non-linear normal force coefficient response in mildly separated flow regime has been achieved. The final network model is able to predict the aerodynamic response to motion histories within the limits of incidence angles and Mach numbers of the training sets. Here, a series of generalisation tests are explored and, for each new test case, the respective time-histories of the separation point, as modelled in the Beddoes model, are shown in Appendix A.

The test in Figure 4.9 for a ramp-up incidence motion history from $-5°$ to $5°$, shows that linear unsteady aerodynamic behaviour is also incorporated in the final FIR network model. Simulations in the range of weakly non-linear behaviour are shown in Figures 4.10 and 4.11, where the FIR network model outputs are able to provide satisfactory predictions of the normal force coefficient response. In addition, the non-linear dependency of the aerodynamic response on Mach number is also reproduced by the FIR network model, since each case is calculated for a Mach number different from the training sets. The good agreement between actual and network outputs is due mainly to characteristics of the Beddoes model data. Indeed,

the use of superposition of indicial responses and a weak separation description based on a Kirchhoff model (*cf.* Appendix A), results in smooth aerodynamic responses that facilitate the training process.

Generalisation tests considering different frequencies in oscillatory motion cases, have shown that the FIR network model is capable of predictions for a wide range of motion frequencies. Figures 4.12 and 4.13 illustrate the lower and higher frequency cases, respectively, compared with the frequency of the oscillatory case for training. Small discrepancies are only observed when the frequency is made four times the frequency of the oscillatory training case.

Three test cases, however, illustrate the network model behaviour for input histories beyond the training limits. Figure 4.14 presents a test case in which a severe incidence motion history is assumed. Starting at a constant incidence of 10°, the angle of attack abruptly changes from ramp-down (reaching $-15°$) to ramp-up in a time interval of 0.3 seconds, returning again to a constant incidence of 10°. During the incidence changes, discontinuous behaviour of the aerodynamic functional is manifested by severe separated flow behaviour as exposed in section 4.2. This feature can also be observed from the evolution of the separation point, as presented in Figure A.16 in Appendix A. Consequently, the FIR neural network response provides a poor approximation to the aerodynamic response over this part of the motion history, but most of the overall unsteady aerodynamic response features can still be reasonably predicted by the model.

Figure 4.15 presents a test case where the incidence motion is within the modelling range, but for a Mach number larger than the values used in the Mach number range for training: that is, $M = 0.55$. The incidence motion is adopted as a ramp-up from an incidence of 0° to 10°. Although the discontinuity induced by increasing the freestream speed (and consequently, separation; *cf.* Figure A.17) is not so strong, the effect on the normal force response functional is sufficient to exclude this case

from the range of applicability of the identified FIR network model. For the last case (*cf.* Figure 4.16) a ramp-up motion to 15° illustrates the network model performance beyond the limits of weakly non-linear behaviour of the aerodynamic response functional. Again, the performance of the FIR neural network model is compromised due to a severe form of non-linear behaviour exhibited in the aerodynamic response. This severe non-linearity, in the form of flow separation, can be confirmed in Figure A.18 in Appendix A.

The unsteady aerodynamic response in the transonic regime presents a more complex case for the identification process. Moreover, two output FIR network models also represent a more complex searching space for the identification process, in terms of supervised training. As a consequence, the training processes require more iterations and difficulties arise in the selection of the training parameters.

For the identified FIR network model at fixed Mach number ($M = 0.65$) the non-linear behaviour of both lift force and pitch moment coefficients are adequately captured. Some discrepancies are associated with the pitch moment coefficient response; however, these are explained by the more severe non-linear characteristics of the pitch moment response. Nevertheless, these discrepancies do not spoil the prediction of the main features of the pitch moment response. For the lift force coefficient good agreement with the training sets is achieved.

Despite the complexity of the searching space, the resulting network architecture is shown to be very simple (*cf.* Figure 4.21). The simplicity of the network architecture, in association with good generalisation results, reinforces the satisfactory performance of the identification process.

Generally, the predictive capabillities of the identified model of the unsteady aerodynamic response in the transonic regime (for fixed $M = 0.65$) are shown to be satisfactory for the majority of the test cases not contained in the training sets. This is particularly true for the lift force coefficient responses. When tested in the linear

range of the unsteady aerodynamic response, the identified FIR network model has not shown good predictive qualities, although the error in the overall responses is not patircularly large. The case, illustrated in Figure 4.23, is obtained from a sinusoidal incidence motion with mean angle of attack equals to zero and 0.5° amplitude at frequency of 10 Hz.

In the non-linear range of the unsteady aerodynamic response, the FIR network model presents good generalisation for low frequency oscillatory cases. Figure 4.24 presents a case where the sinusoidal incidence motion is half the frequency of the corresponding case used in the training sets. Only a few discrepancies can be observed in the pitch moment response of the FIR network model. Nevertheless, when a higher frequency sinusoidal motion is tested, as in the case illustrated in Figure 4.25, the discrepancies in the pitch moment response increase. Here, the frequency of the incidence motion is twice that of the corresponding case used in the training sets.

Figures 4.26 to 4.30 present more generalisation tests of the FIR neural network model of unsteady aerodynamic response in the transonic regime (fixed $M = 0.65$). The cases correspond to pulses, ramps, and oscillatory input motions in the range of non-linear behaviour of unsteady aerodynamic response. The FIR network model outputs for those cases indicate good agreement with the desired unsteady aerodynamic responses. Significant discrepancies arise in the pitch moment responses and for the cases in Figures 4.28 and 4.29, the steady-state behaviour of pitch moment response in the presence of shock waves (*cf.* Figures B.13 and B.14) shows major disagreement with the Euler CFD code outputs.

Identification of a FIR neural network model of the unsteady aerodynamic response is considered for a range of Mach numbers in the transonic regime. This problem presents a greater challenge to the identification process. The increased size of the search space, in comparison with the previous cases, requires a greater

number of training iterations. These difficulties are also observed in the complexity of the final adapted architecture, as depicted in Figure 4.40. Although disprepancies in the training results for the pitch moment are larger than in the previous model for fixed Mach number, the overall features of the unsteady aerodynamic response is captured. Again, the lift force coefficient response of the FIR network model is better than the pitch moment response.

The test case in Figure 4.42 considers an incidence motion history compatible with unsteady aerodynamic response in the linear range. An oscillatory motion with similar frequency to the oscillatory cases in training sets. amplitude of 0.5°. and Mach number equal to 0.7, is considered. The FIR network model output for this case reveals a reasonable approximation of the lift coefficient response; however. a very poor prediction of the moment coefficient response, in the form of large out of phase motion, can also be observed.

In futher cases (*cf.* Figures 4.43 and 4.44), the FIR network model is tested for oscillatory motions at different frequency values compared with the oscillatory cases in training sets. The frequency for the case in Figure 4.43 is twice that of the training set oscillatory case, the amplitude is 4.5° and the Mach number is equal to 0.625. The second case in Figure 4.44 presents a frequency value that is half that of the training set oscillatory case, the amplitude is 1.5° and Mach number is 0.725. The FIR network approximation of the pitch moment coefficient response for these cases, again, exhibits very poor results. The poor approximation properties are due to poor training results for the pitch moment output as observed from Figures 4.31 to 4.39.

Figures 4.45 to 4.47 present test cases at Mach number equal to 0.65. with the purpose of comparing the FIR network model response with the FIR network model obtained for a fixed Mach number (*cf.* Section 4.4.1), as shown in Figures 4.26 to 4.28. Most of the differences arise in the pitch moment responses where large

discrepancies can be observed. Despite the differences and the prediction disparities, the FIR neural network model for a range of Mach numbers in the transonic regime, maintains the principal features of the unsteady responses, even for Mach numbers different from those in the training sets. Larger errors in the steady-state response of the pitch moment in the FIR network model can be observed for a range of Mach numbers in which shock waves are present.

Further test cases with different incidence motion histories and Mach numbers are presented in Figures 4.48 to 4.52. These tests illustrate how the FIR network model is able to predict the unsteady aerodynamic responses within the limitations imposed by the training sets.

Finally, the behaviour of the FIR network model is tested for a case with Mach number beyond the training limits. An oscillatory case with the same frequency as that of the oscillatory case used in the training sets, amplitude 1° and Mach number of 0.75, is considered. Here, the flowfield is characterised by the presence of shocks whatever the incidence angle, as observed in Figure B.28 in Appendix B. Figure 4.53 shows that the FIR network model for this case and in particular, the lift force coefficient response, provides reasonable predictive characteristics, despite the fact that the Mach number ($M = 0.75$) is beyond the limits of the training sets.

4.6 Summary

This Chapter has illustrated the application of multi-layer functionals, as realised by FIR neural networks, for the identification of non-linear unsteady aerodynamic response models. The identification process, defined in terms of a genetic search algorithm, has been shown to be satisfactory for the production of robust FIR

network models of non-linear unsteady aerodynamic response.

Two classes of flow regimes are considered for the identification of the FIR network models of unsteady aerodynamic response. Both regimes are presumed to be free of flow effects that may lead to discontinuities in the functional representation. The first case considers the weakly non-linear behaviour of the unsteady normal force response in mildly separated flowfields in response to variations in the incidence motion of a two-dimensional NACA 0012 airfoil for a range of subsonic Mach numbers. The database for identification is obtained from a semi-empirical Beddoes model (Niven & Galbraith, 1991). The FIR neural network model for the weakly non-linear unsteady normal force response in mildly separated flows has provided good generalisation within the range of incidence motions and Mach numbers used in the training. The FIR network model has also been validated for a range of incidence motion frequencies; however, when tested for cases where severe separated flow conditions are present, the predicted results of the FIR network model show major discrepancies.

For the second flow regime, the non-linearities in the unsteady aerodynamic responses, caused by compressibility effects in the transonic regime, are considered. The database is obtained using a CFD code (Dubuc *et al.*, 1997) based on the solution of the Euler equations. Two FIR network models of the unsteady aerodynamic response are identified. The first one identifies the unsteady lift force and pitch moment responses to variations in the incidence motion of a two-dimensional NACA 0012 airfoil at a fixed Mach number ($M = 0.65$), while the second model is for a range of transonic Mach numbers. The training results for both transonic cases show better approximation for the lift force response in comparison with the pitch response. The same observation is made for the generalisation cases. The discrepancies in the unsteady pitch moment are larger in the FIR network model applicable to a range of transonic Mach numbers, when compared with the same

cases for the FIR network model value at a fixed Mach number.

Generally, the training process has shown satisfactory performance. The complexity of the non-linear input-output mapping for the unsteady transonic cases has resulted in to a longer training process. The division of the training process into stages has also been used as a means of improving the convergence performance during the process.

Chapter 5

Conclusions and Recommendations for Further Research

5.1 General Conclusions

Multi-layer functionals furnish appropriate representation of the non-linear motion-induced unsteady aerodynamic response. Finite impulse response (FIR) neural networks are practical realisations of multi-layer functionals and can be used to accurately approximate the non-linear unsteady aerodynamic response. The FIR neural network model can incorporate widely different physical behaviour of the non-linear unsteady aerodynamic response. Identification of an appropriate FIR neural network, that provides a parametric input-output model, is achieved via a supervised training process using multiple sets of motion-induced unsteady aerodynamic responses.

The advantages associated with multi-layer functional approximation of non-linear unsteady aerodynamic response via FIR neural networks are:

(i) the ability to account for non-linearities and time-history dependencies en-

countered in unsteady flow regimes:

(ii) the implementation of parametric multiple input multiple output models for aeroelastic applications, allowing fast evaluation of the aerodynamic responses;

(iii) static parameters, such as Reynolds number or Mach number, can be used as inputs to the neural model, thereby increasing the range of flow conditions and, as a consequence, the applicability of the neural model;

(iv) the difficulties related to the traditional non-linear system identification approaches are diminished.

The identification procedure is based on genetic search to optimise the FIR network architecture, combined with a simplification of the simulated annealing algorithm to update weight and bias values, and to improve the convengence rate of the process. This approach is shown to overcome many of the difficulties associated with the standard temporal back-propagation algorithm, and also to provide a more flexible manipulation of the FIR network architectures; for example, by allowing each network connection to have a different time-delay value. Therefore, the combination of genetic algorithms and simulated annealing provides an appropriate framework for training temporal of neural networks.

The training performance is shown to require long processing time. This can be attributed to the combination of the genetic algorithm for the FIR network architecture optimisation and the methodology of updating the weight and bias values in the networks. The rate at which the genetic search optimises network architectures tends to be higher than that at which the perturbation process updates weight and bias values in the respective FIR neural networks. Numerical tests are required to establish a good set of training parameters to obtain an acceptable convergence rate of the overall process. However, differences in the efficiencies of

the various elements involved in the training process have not compromised the identification of the unsteady aerodynamic response functionals.

A limitation associated with the training process is related to the size of the maximum network architecture. For large values of time-delays per connection and number of neurons, the encoded FIR network results in a lengthy chromosome that requires more computational effort.

Application of the identification process for two different types of unsteady flow regimes, where the flow effects induce non-linear behaviour to the unsteady aerodynamic responses, shows encoraging results to establish the FIR neural network modellling as an important framework for systematic production of unsteady aerodynamic models. The first flow regime considers the weakly non-linearities in unsteady aerodynamic responses for mildly separated flowfields, while the second one considers the non-linear behaviour of unsteady aerodynamic response in the transonic regime shock waves and their dynamic excursion occurs.

The generalisation test results show that, given only limited training set data, the FIR neural network models are capable of accurate predictions of the non-linear unsteady aerodynamic responses due to any motion history and flow parameter within the training boundaries.

For the multi-layer functional representation of the weakly non-linear unsteady force response in mildly separated flow regimes, generalisations have revealed good approximation properties in the range of Mach numbers, motion histories and frequencies considered. A major limitation of the FIR neural network model is the restriction to continuous functionals, since simulations considering highly separated flows have shown larger errors with respect to the Beddoes model outputs to the unsteady aerodynamic responses in the motion segments with strong separation effects.

The results for the non-linear unsteady aerodynamic responses in the transonic

regime show that the multi-layer functional in the form of a FIR neural network represents an efficient model form. Although the results for the moment coefficient response were poor in both cases, good predictions of the lift force coefficient response within the training range of motion histories and Mach numbers are achieved. Evidently, this reflects imperfections in the training process in terms of avoiding local minima. Nevertheless, the complexity of non-linearities involved in these aerodynamic cases, added to the limitations on information contained in the training sets available for the identification process, represent important factors in the final FIR network model. The influence of the training sets on the identification process can be observed in the case at fixed transonic Mach number. For the restriction to fixed Mach number, the FIR network model is able to produce reasonably good predictions even for the moment coefficient responses. In contrast, when the problem is expanded for a range of transonic Mach numbers, the complexity of the input-output mapping suggests that more training cases, and different training conditions, could improve the final outcome.

5.2 Topics for Future Work

To extend the capabilities of the multi-layer functional representation approach to non-linear unsteady aerodynamics, improvements to the conventional operators incorporated in the identification algorithm are necessary. However, the combination of genetic algorithms and neural networks shows to be flexible enough to provide a large source of possibilities for identification schemes. Therefore, modifications to chromosome description, the use of different (and possibly more efficient) genetic operators, and the definition of other cost function forms for the optimisation of the

network architecture, require further investigation.

For the particular problem of the determination of weight and bias values, a proper application of the simulated annealing algorithm is a good alternative. The great difficulty in this case is the prescription of a suitable cooling schedule and a reasonable temperature definition. Moroever, as the number of variables required to be optimised (weight and bias values in this case) is generally large in a FIR neural network, random perturbations can be compromised. To help in these issues, the work by Otten & van Ginneken [1989] may provide a valuable contribution.

The temporal back-propagation algorithm (Wan, 1990a, 1990b) can also be combined with the genetic algorithm as a means of updating weight and bias values. However, an immediate drawback associated with this approach is a loss in the FIR neural network architecture's ability to possess different time-delays per connection. Apparently, no reduction in the number of training steps or in the processing time would be achieved, but an advantage of this approach could be in terms of better relative convergence rates between architecture and weight optimisation.

Despite the achievements of the multi-layer functional representation in discrete-time, continuous-time versions are desired to overcome some of the limitations of the discrete version; for example, sampling and implementation for general problems in aeroelastic analysis and control. A possible generalisation of the algorithm to accommodate continuous-time models could be achieved by using the temporal neural network approach and respective continuous-time temporal back-propagation algorithm, developed by Day & Camponese [1991] and Day & Davenport [1993].

Another important issue for future work in multi-layer functional approximation of non-linear unsteady aerodynamics is the production of representaions that can account for discontinuous behaviour; for example, as in separated flows. Tobak & Chapman [1985] have already discussed an alternative functional model based on the composition of continuous functionals for the various stable unsteady aerody-

namic responses resulting from aerodynamic bifurcations. Using the same concept, multi-layer functional approximate models could be composed to produce non-linear unsteady aerodynamic response models in separated flowfields. Moreover, numerical techniques such as *fuzzy logic* (Yager & Filev, 1994) could also be combined to the multi-layer functional representation to produce a rationale for the discontinuous behaviour of the non-linear unsteady aerodynamic response.

Traditional methods for non-linear dynamic systems identification; for instance, Volterra and Wiener functional series and block-oriented models (Billings, 1980), also represent good tools for non-linear unsteady aerodynamic modelling. However, the limitations imposed by the kernel determination and the need for special forms of inputs (for example, Gaussian noise inputs are used in the Wiener functional series forms and block-oriented realisations) have contributed to discourage further application in non-linear unsteady aerodynamic modelling.

However, in the future, new studies on kernel determination may provide the necessary mathematical tools to facilitate the traditional non-linear identification approaches to non-linear unsteady aerodynamic modelling. An example of this is given in Wray & Green [1994], where a special neural network architecture is shown to be equivalent to the finite memory, discrete, Volterra series. The Volterra kernels are then calculated from the internal network parameters by means of network training via the conventional back-propagation algorithm.

Based on the features of Volterra series, one can infer that the same approach (Wray & Green, 1994) could be used to model the continuous non-linear behaviour of the unsteady aerodynamic response, allowing advantages in both aeroelastic analysis and control design problems. Moreover, the extracted Volterra kernels can be easily associated with a bilinear system, that represents a suitable form for non-linear control.

Application of the multi-layer functional representation in aeroelastic modelling

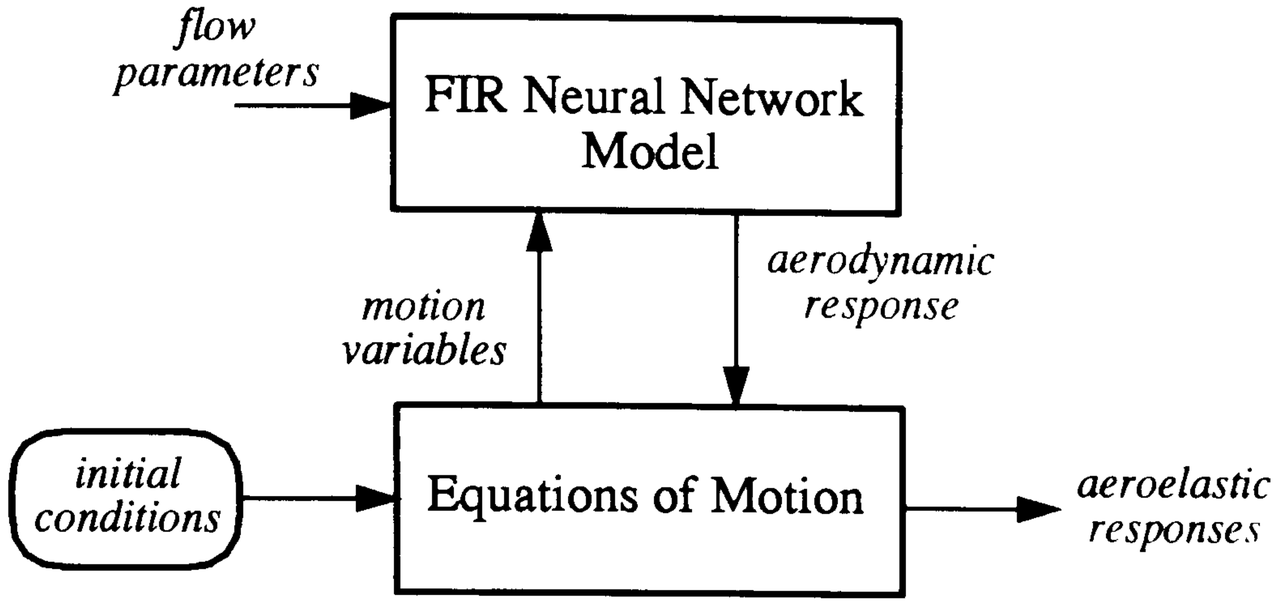


Figure 5.1: Example of utilisation of a FIR neural network model of the unsteady aerodynamic response in an aeroelastic model.

can be considered in many forms can be considered. For example, in Figure 5.1, an aeroelastic model, including a FIR neural network model of the non-linear unsteady aerodynamic response, is illustrated.

In this case the FIR neural network model would use the motion variable values at a particular time instant (accounting for the past motion history) furnished by the equations of motion, to return the respective unsteady aerodynamic responses at the respective time instant. The nature of the discrete-time equations of motion could be linear or non-linear. The design of controllers of the aeroelastic response could also benefit from this type of model. By considering the example in Figure 5.1 as the plant, any conventional linear or non-linear control approach or new approaches; such as, neural networks and fuzzy logic, could be used.

Appendix A

Representation of Unsteady Trailing Edge Separation

The Kirchhoff flow model is among the theories for two-dimensional flows that encompass separated regions. The model provides an approximation of the steady aerodynamic response in terms of a non-linear function of the separation point, f, as illustrated in Figure A.1.

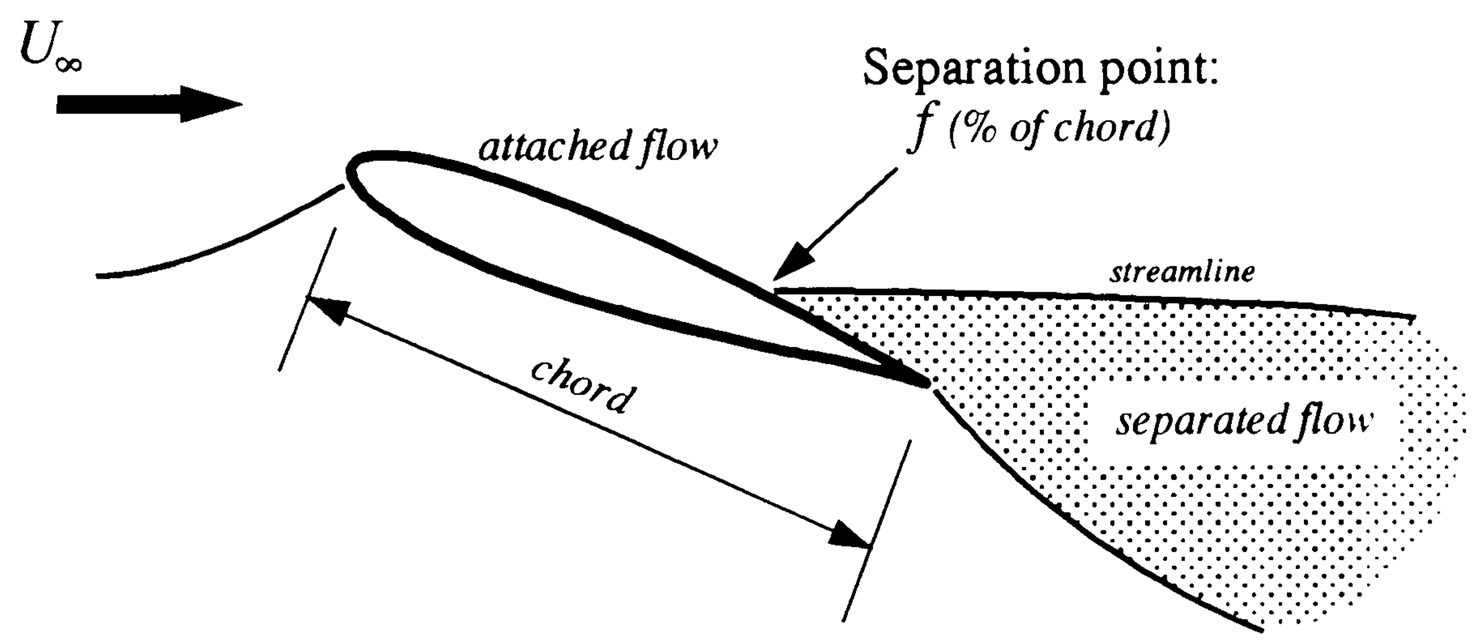


Figure A.1: Kirchhoff Flow Model.

In the Beddoes model (Niven & Galbraith, 1991; Beddoes, 1976, 1982a, 1982b and Leishman & Beddoes, 1986), the Kirchhoff model is adapted for the unsteady aerodynamic case to obtain the variations in the aerodynamic force and moment

with the incidence motion by approximating the relationship between the separation point and motion as a function of the angle of attack. This approach provides non-linear scaling to the unsteady aerodynamic response, until the breakdown point, that represents severe separated flow (stall).

In this Appendix, the separation point histories are presented for each case used in Section 4.3 for training and generalisation tests of the FIR network model of weakly non-linear unsteady aerodynamic response in mildly separated flow. All cases are based on the parameters presented in the Figure 4.3, where the trailing edge is considered here at 100% chord length.

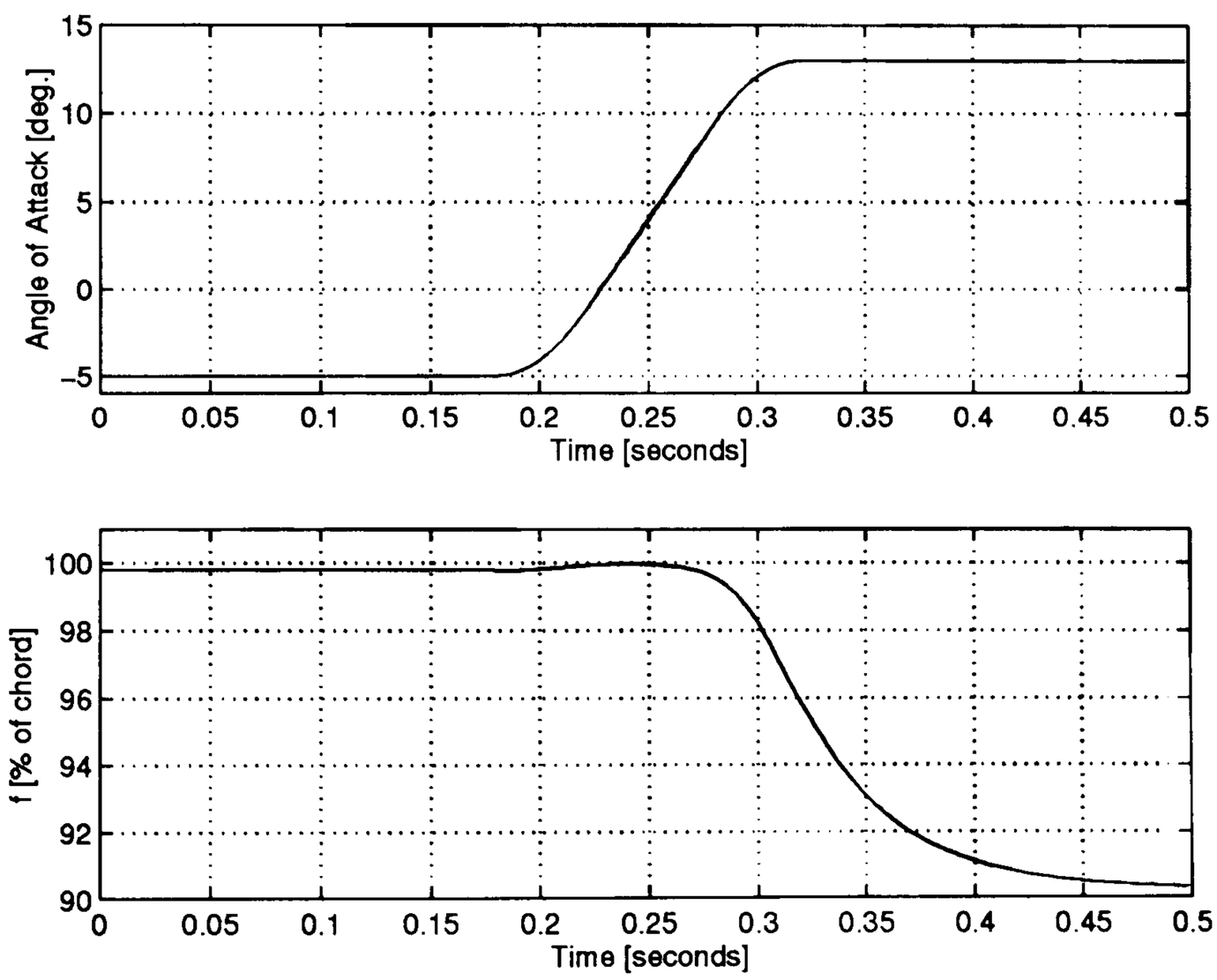


Figure A.2: Separation point excursion in relation to airfoil motion: Case in Figure 4.4 ($M = 0.15$).

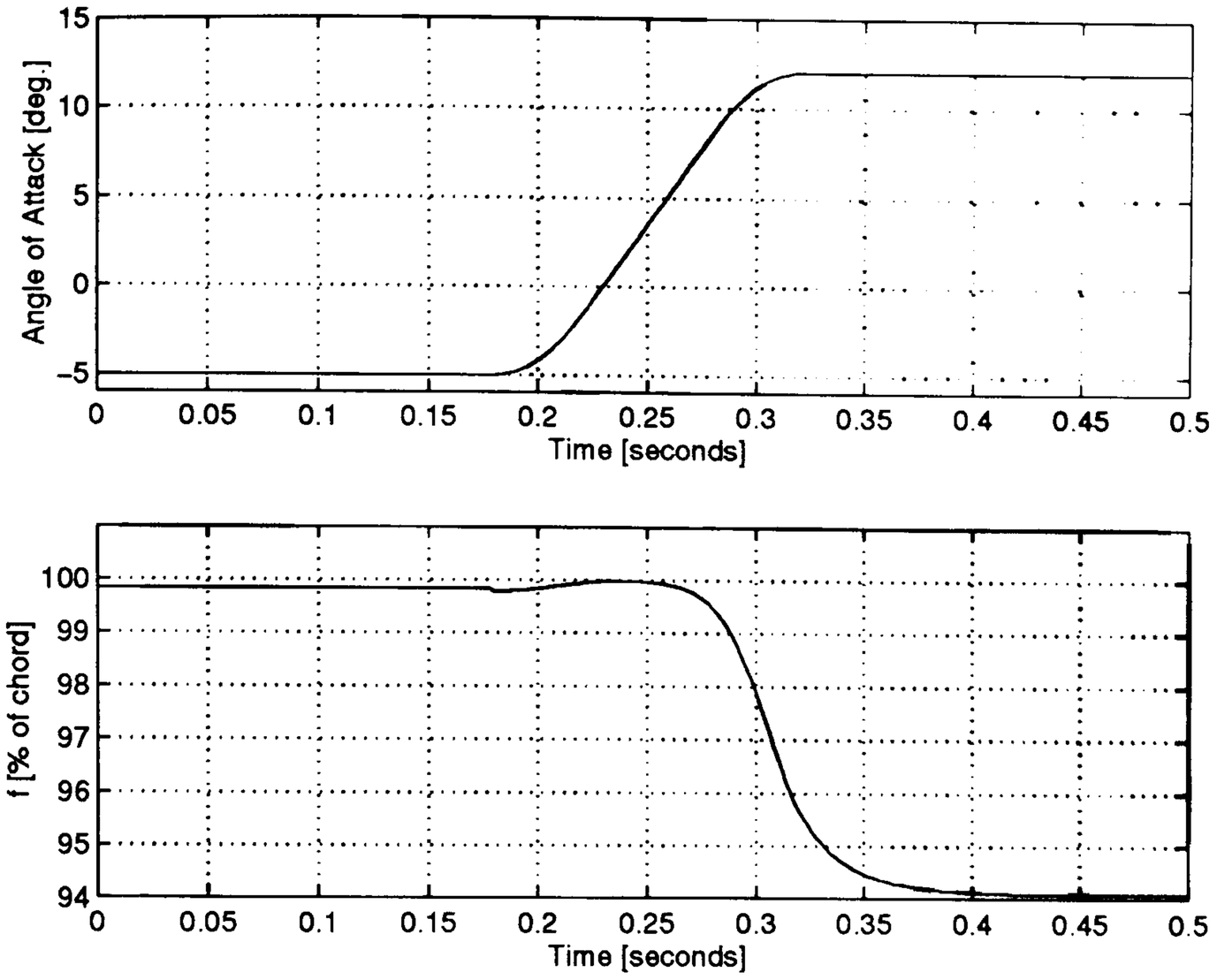


Figure A.3: Separation point excursion in relation to airfoil motion: Case in Figure 4.4 ($M = 0.30$).

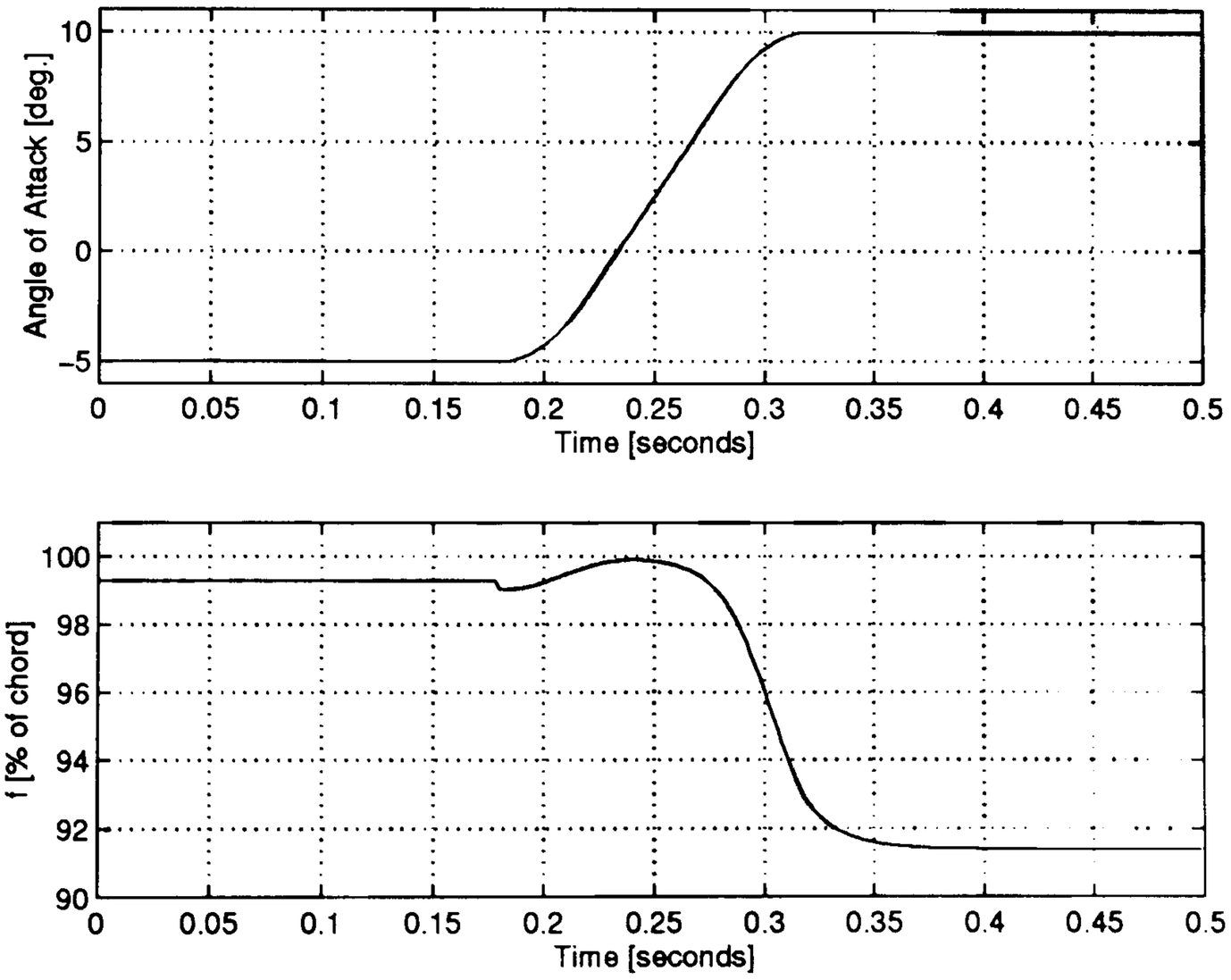


Figure A.4: Separation point excursion in relation to airfoil motion: Case in Figure 4.4 ($M = 0.45$).

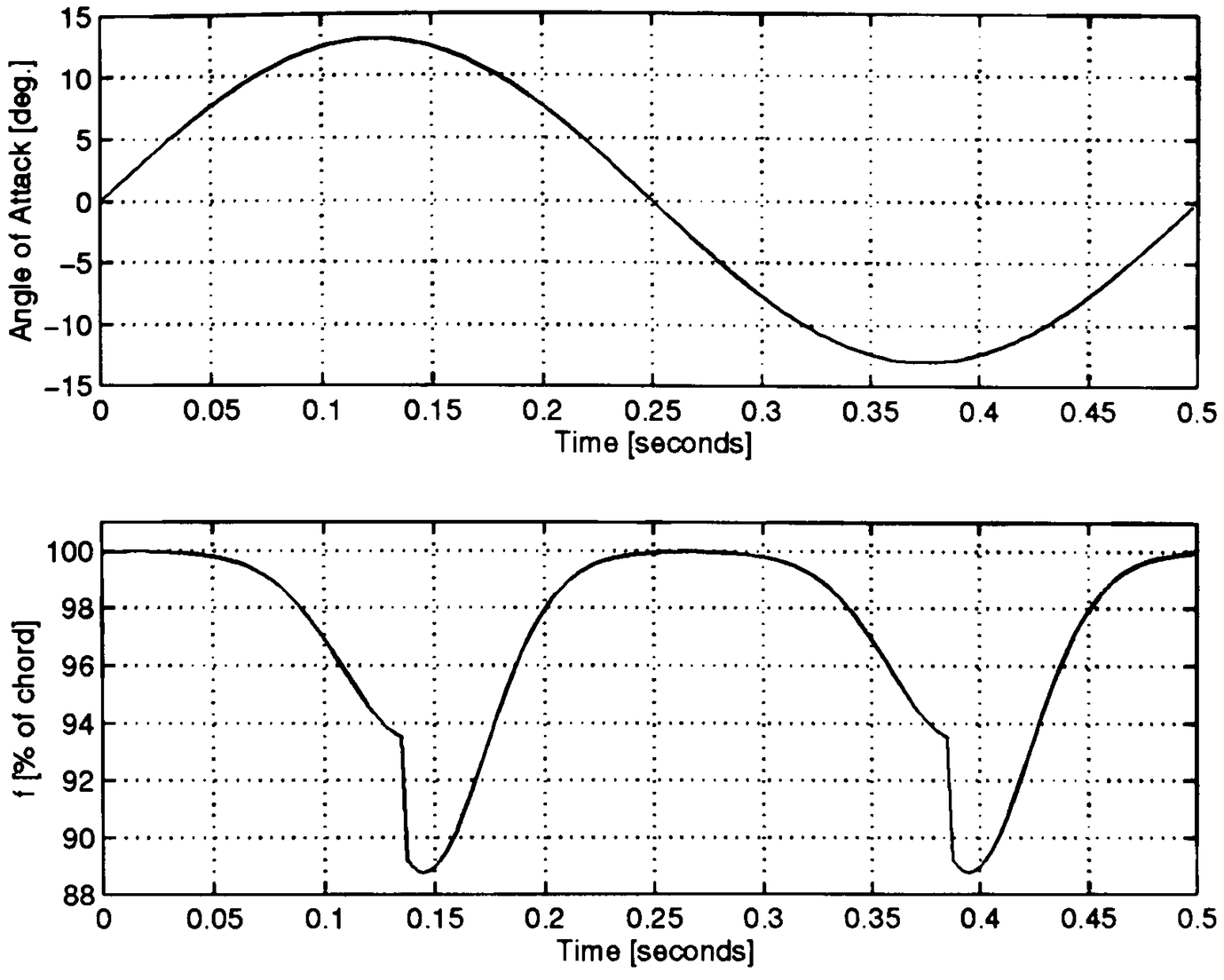


Figure A.5: Separation point excursion in relation to airfoil motion: Case in Figure 4.5 ($M = 0.15$).

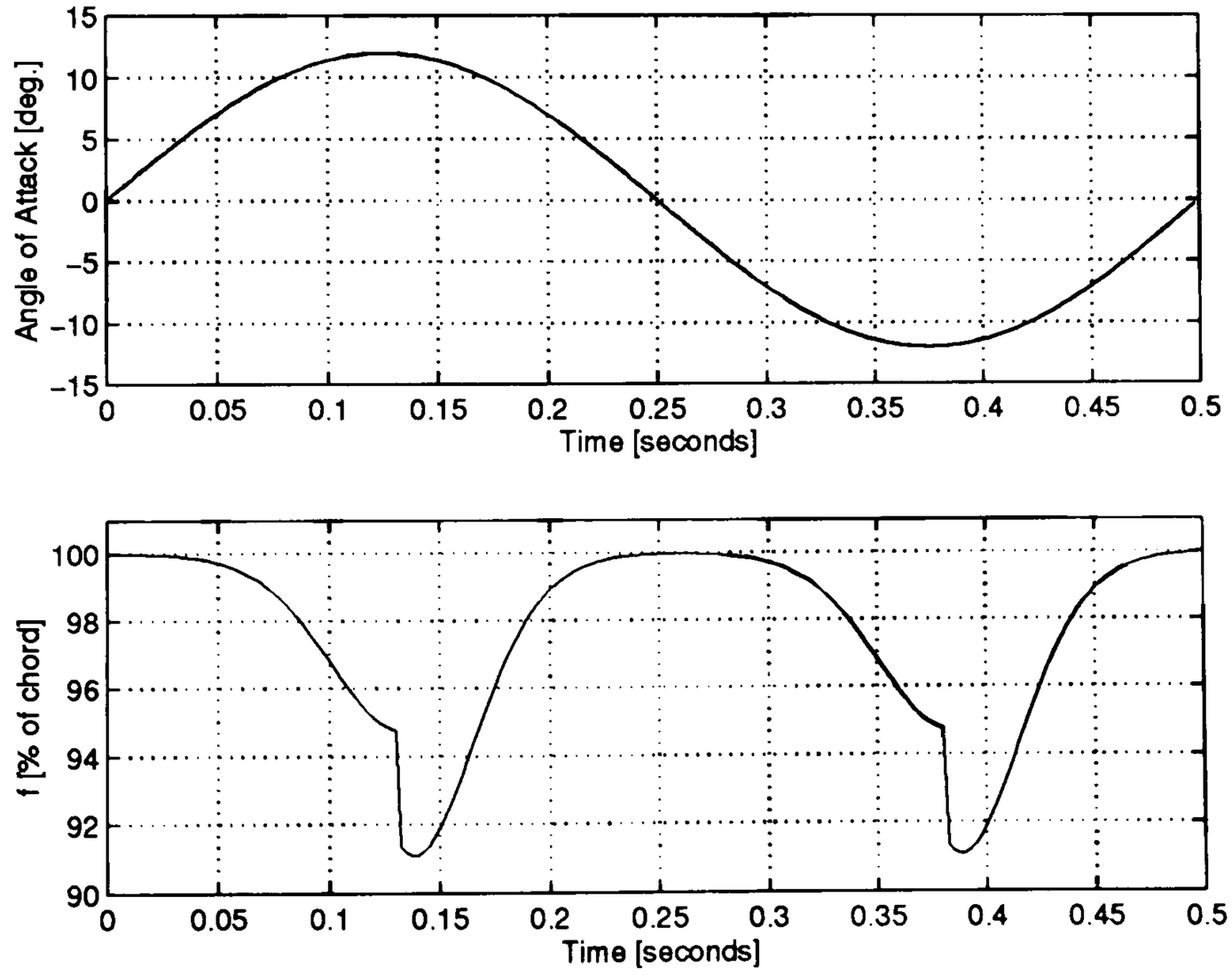
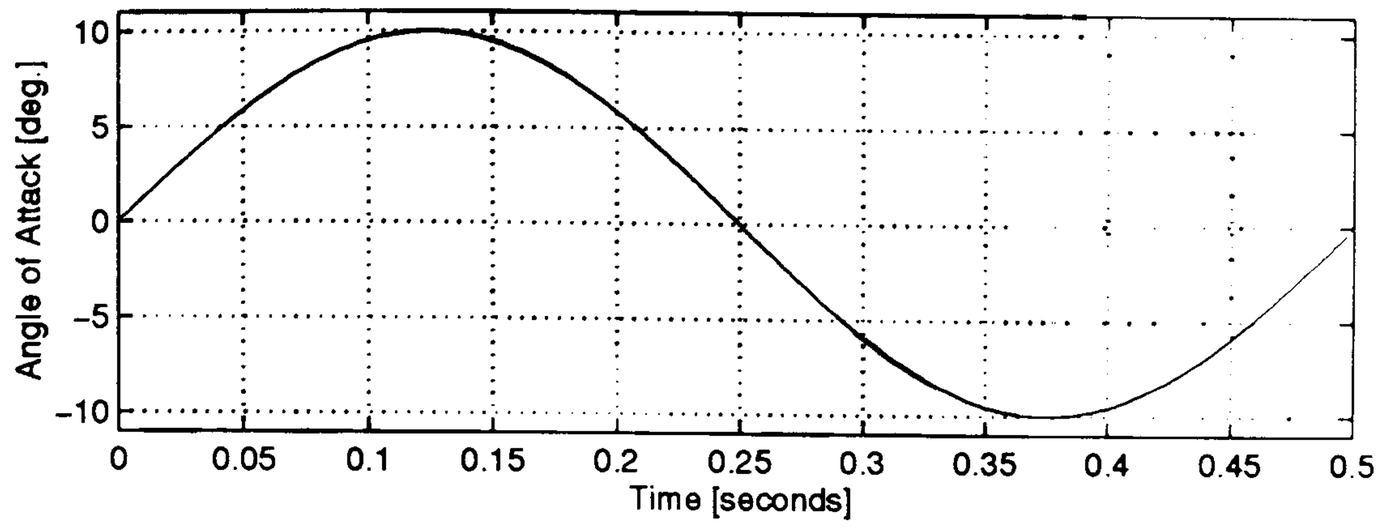


Figure A.6: Separation point excursion in relation to airfoil motion: Case in Figure 4.5 ($M = 0.30$).



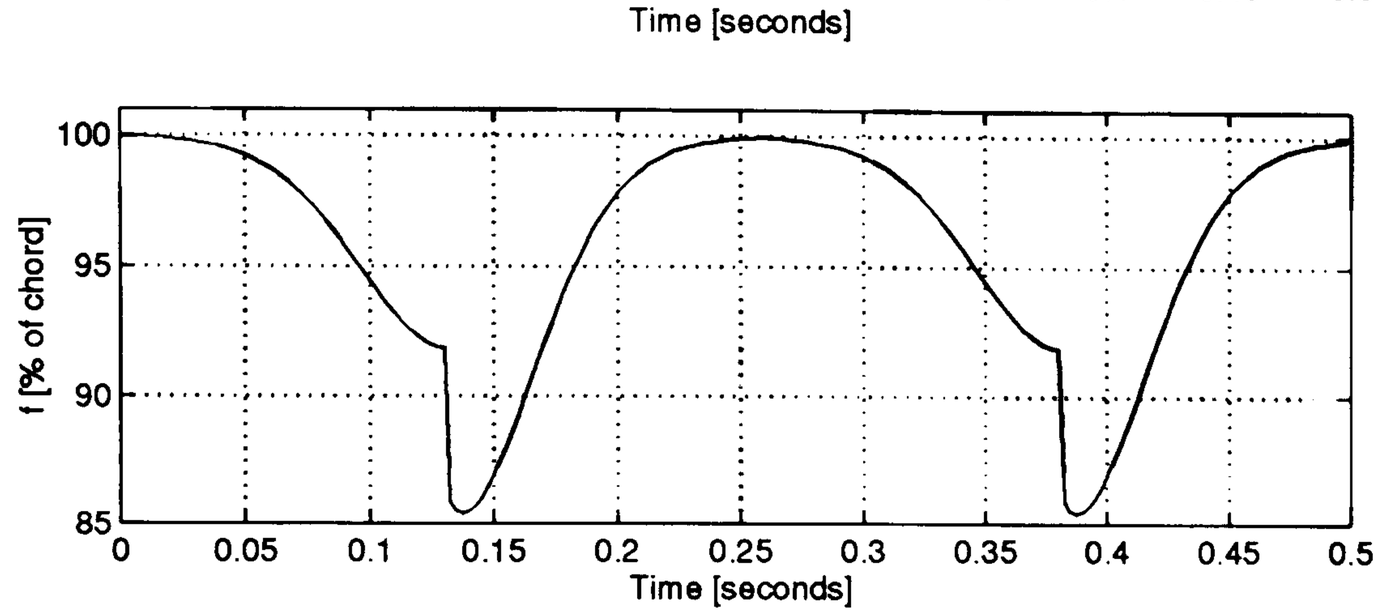
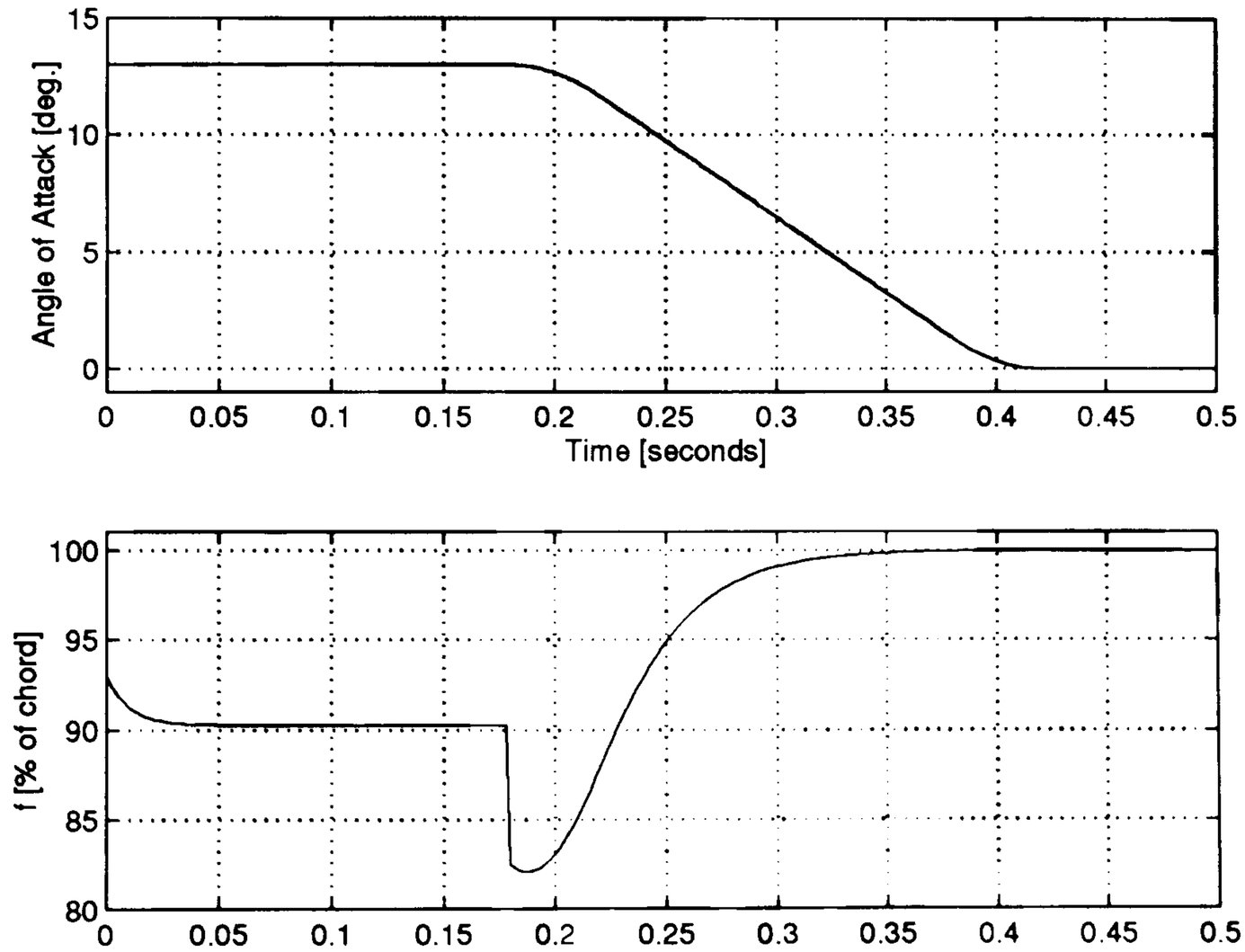


Figure A.7: Separation point excursion in relation to airfoil motion: Case in Figure 4.5 ($M = 0.45$).



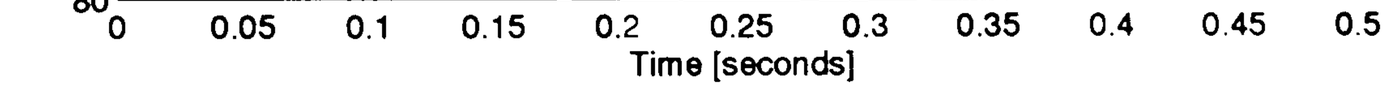
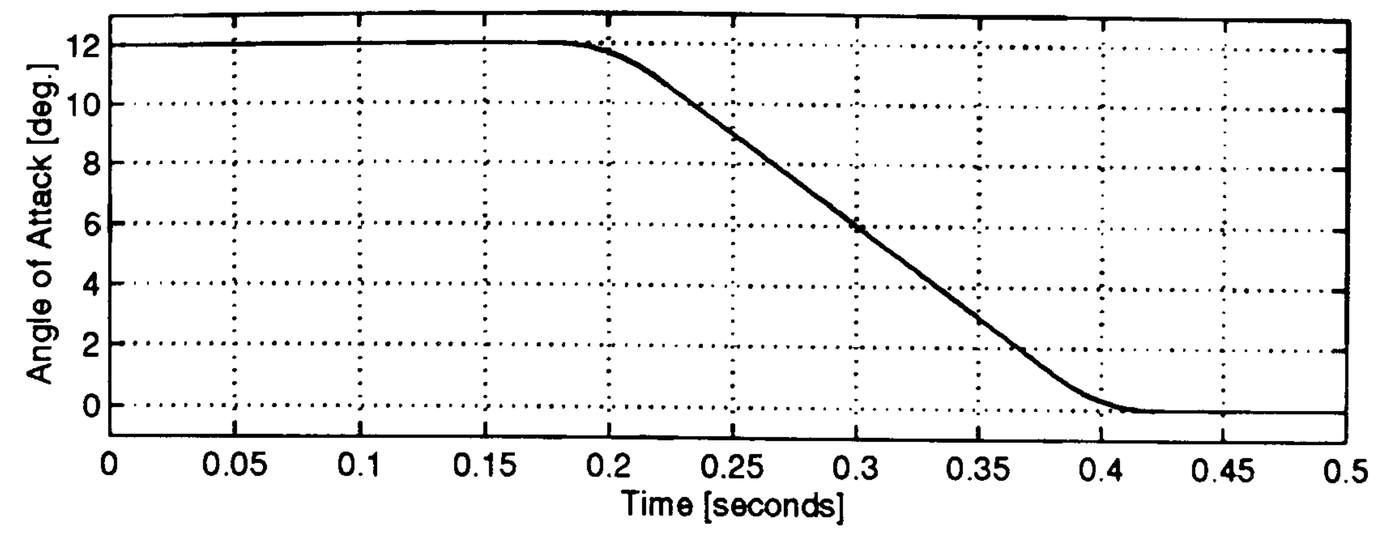


Figure A.8: Separation point excursion in relation to airfoil motion: Case in Figure 4.6 ($M = 0.15$).



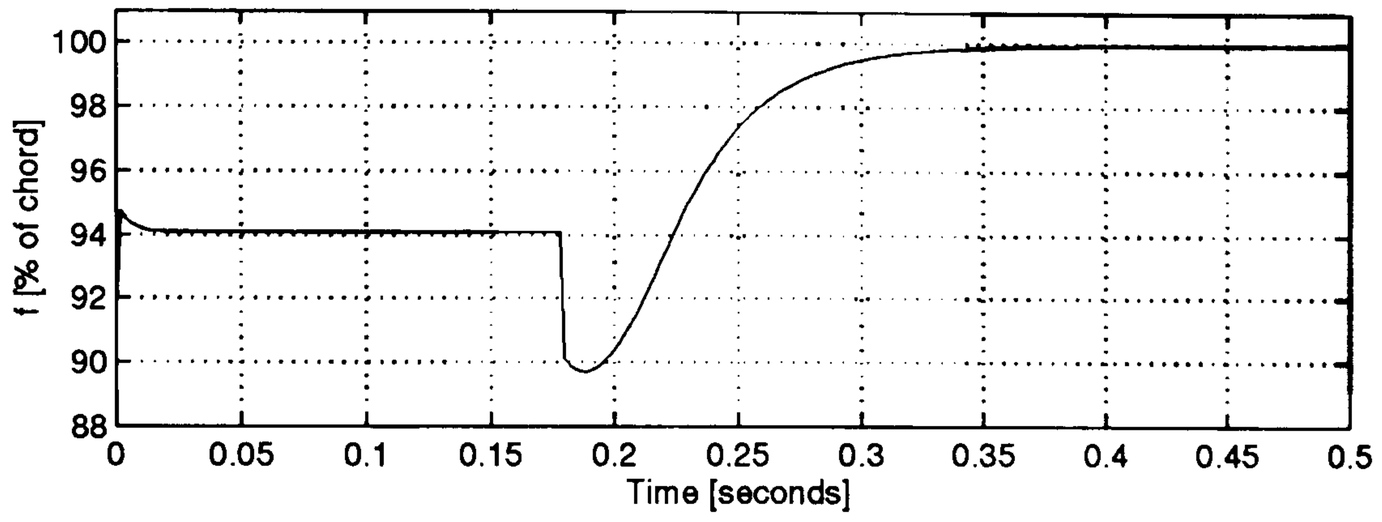
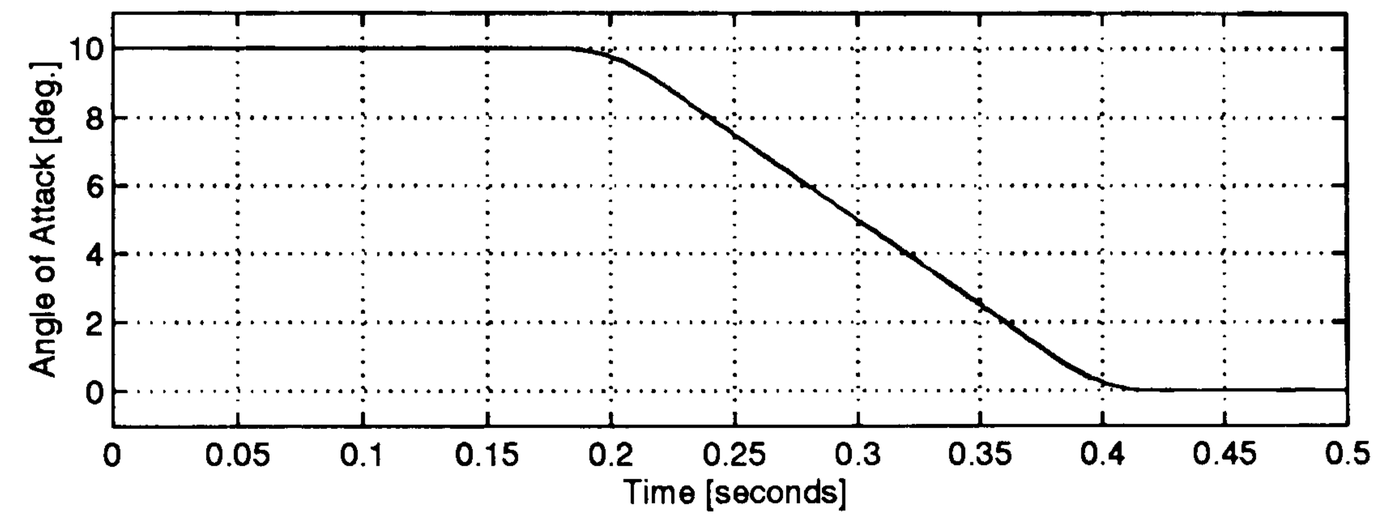


Figure A.9: Separation point excursion in relation to airfoil motion: Case in Figure 4.6 ($M = 0.30$).



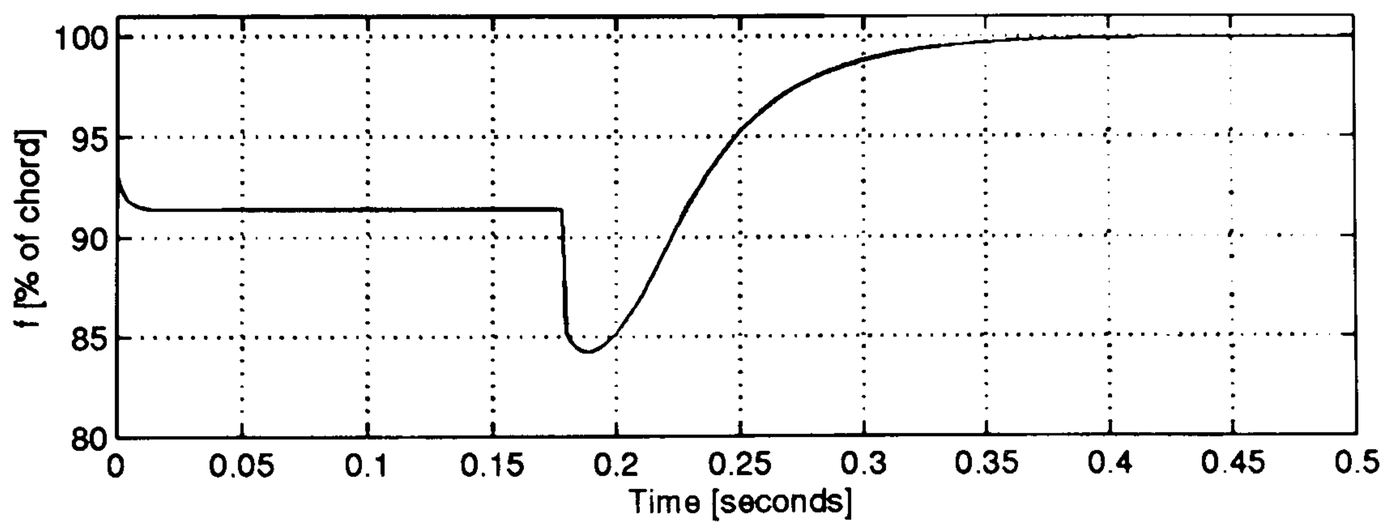


Figure A.10: Separation point excursion in relation to airfoil motion: Case in Figure 4.6 ($M = 0.45$).

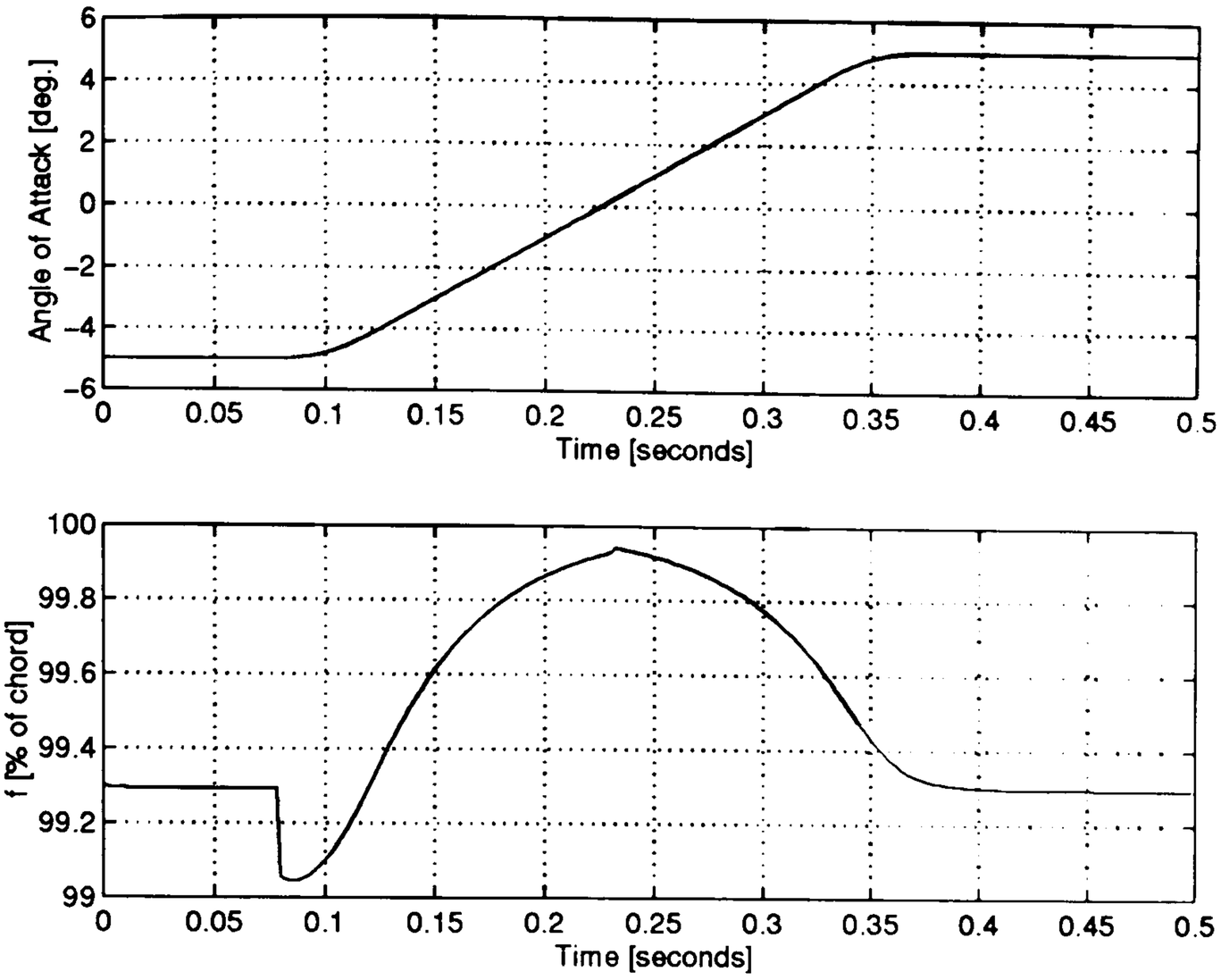


Figure A.11: Separation point excursion in relation to airfoil motion: Case in Figure 4.9 ($M = 0.40$).

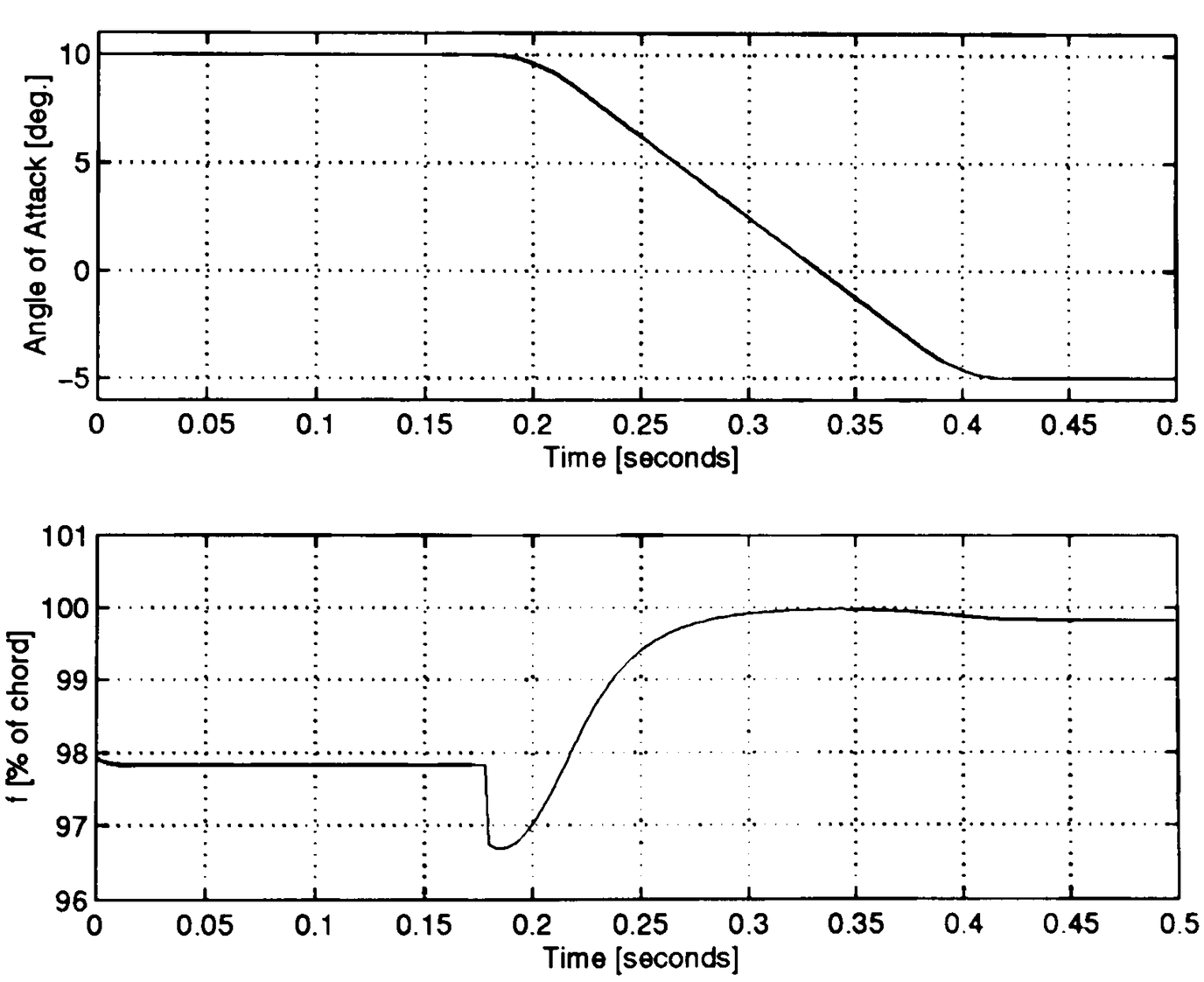


Figure A.12: Separation point excursion in relation to airfoil motion: Case in Figure 4.10 ($M = 0.35$).

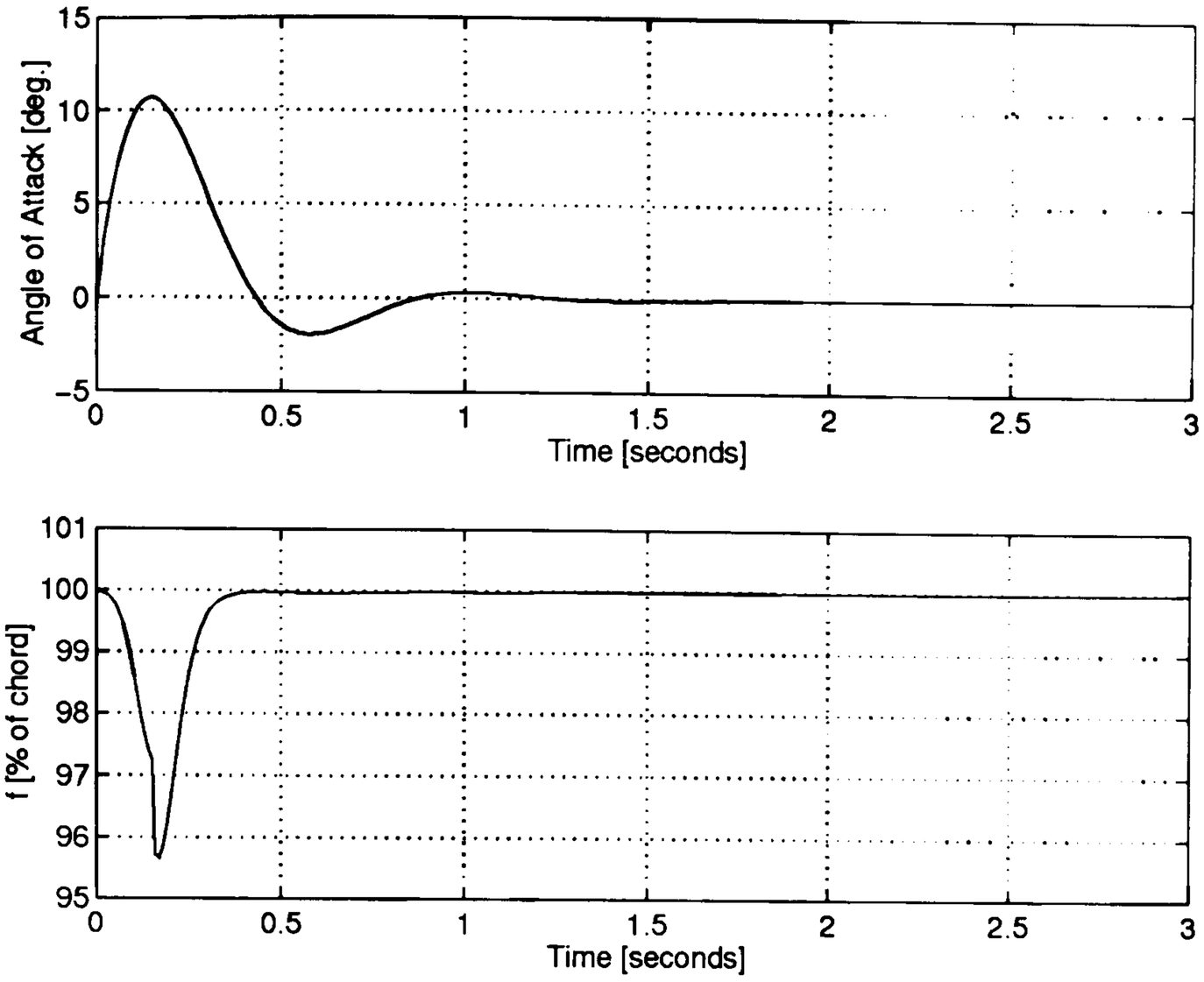


Figure A.13: Separation point excursion in relation to airfoil motion: Case in Figure 4.11 ($M = 0.20$).

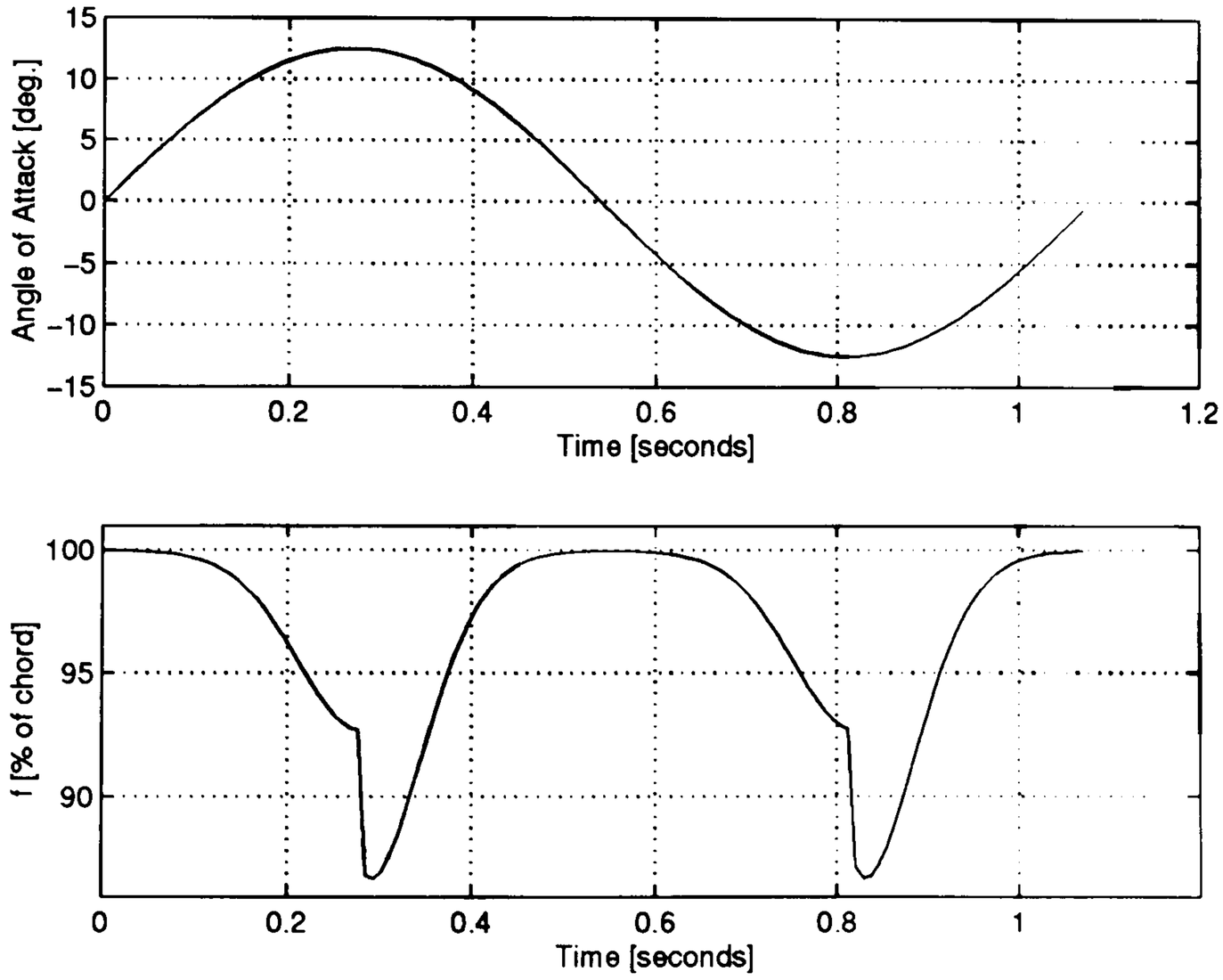


Figure A.14: Separation point excursion in relation to airfoil motion: Case in Figure 4.12 ($M = 0.25$).

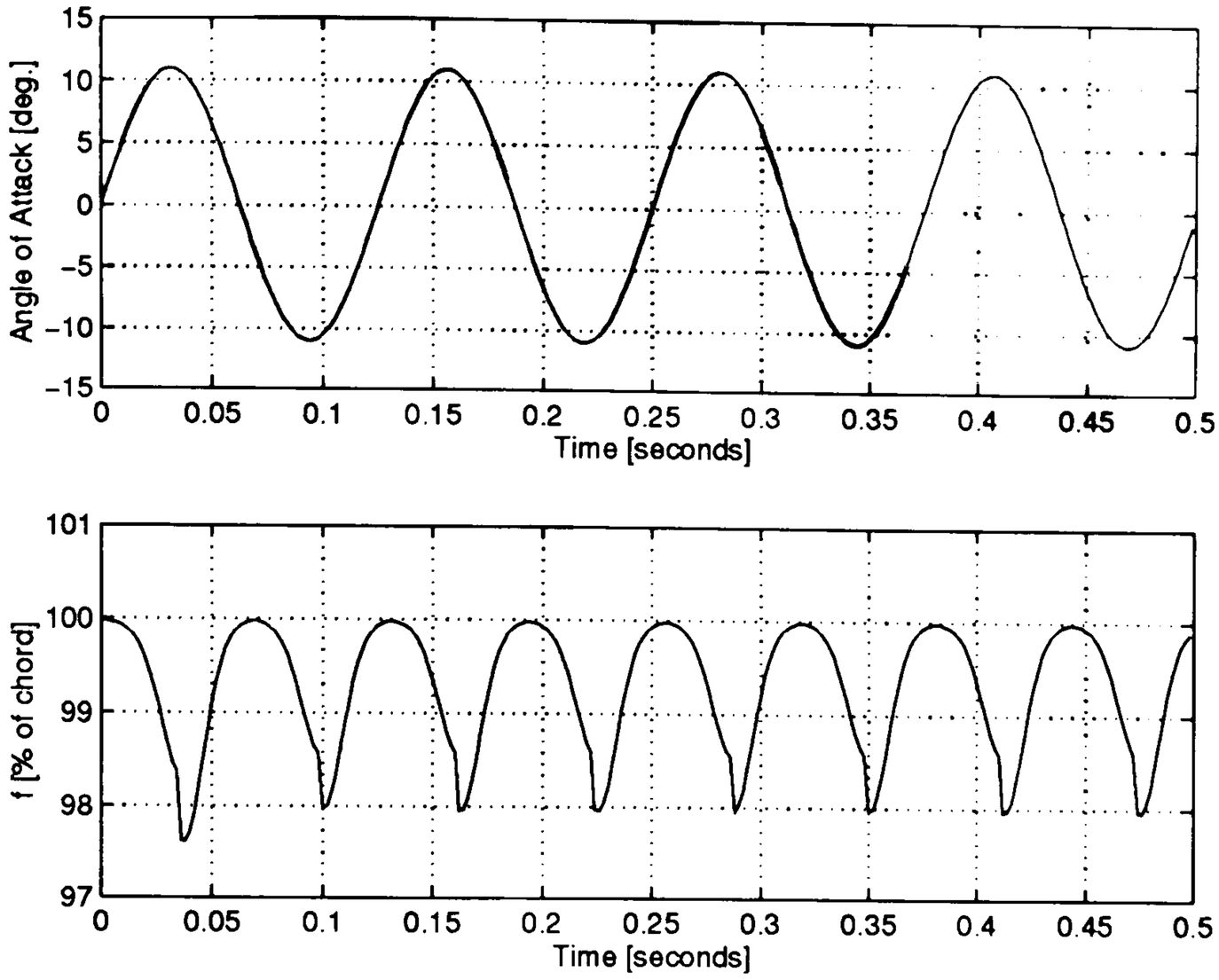


Figure A.15: Separation point excursion in relation to airfoil motion: Case in Figure 4.13 ($M = 0.33$).

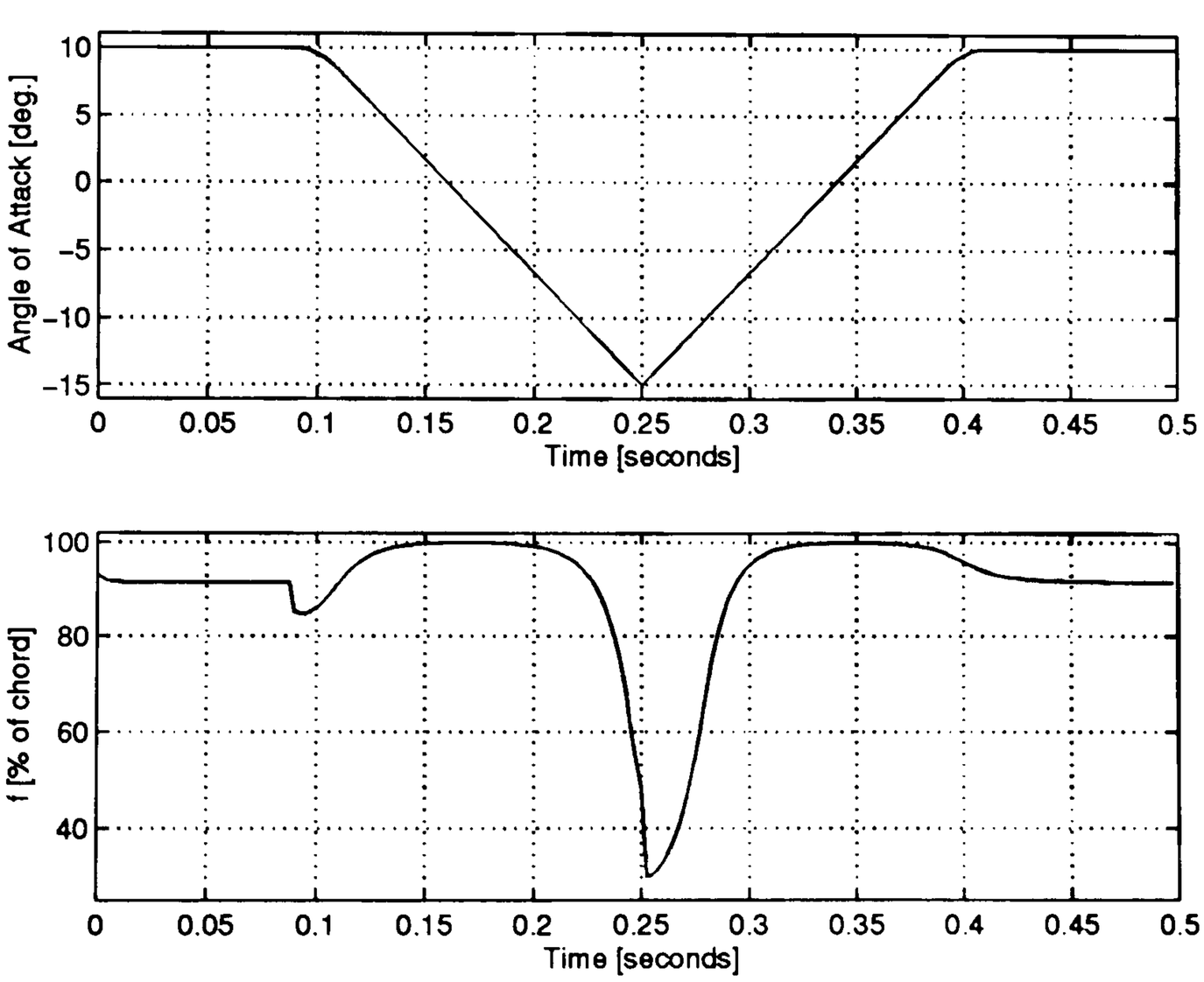
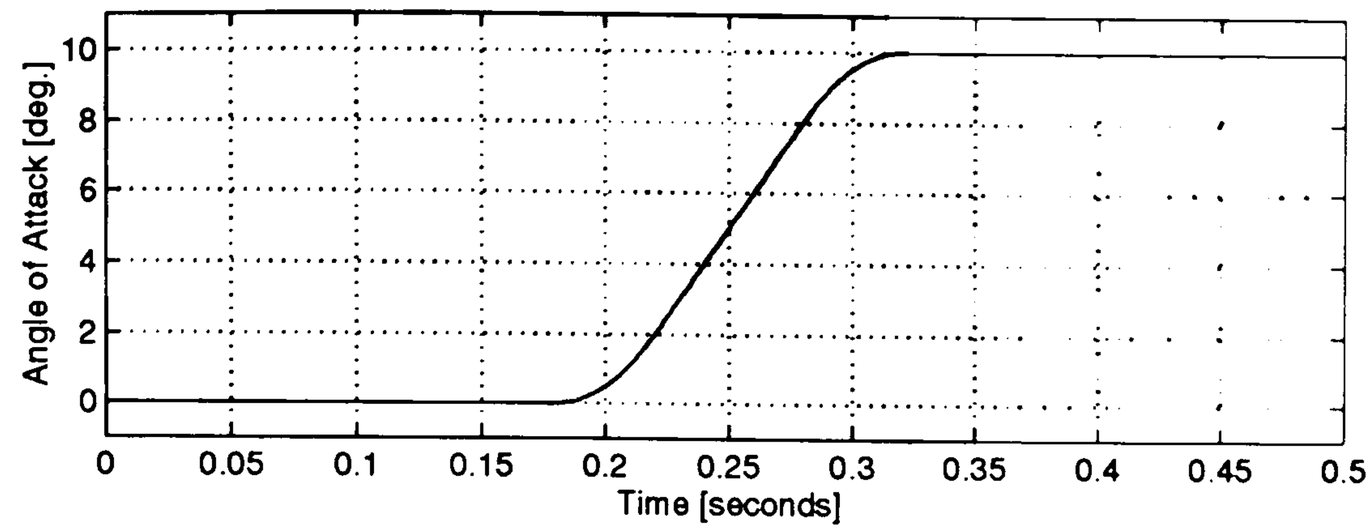


Figure A.16: Separation point excursion in relation to airfoil motion: Case in Figure 4.14 ($M = 0.43$).



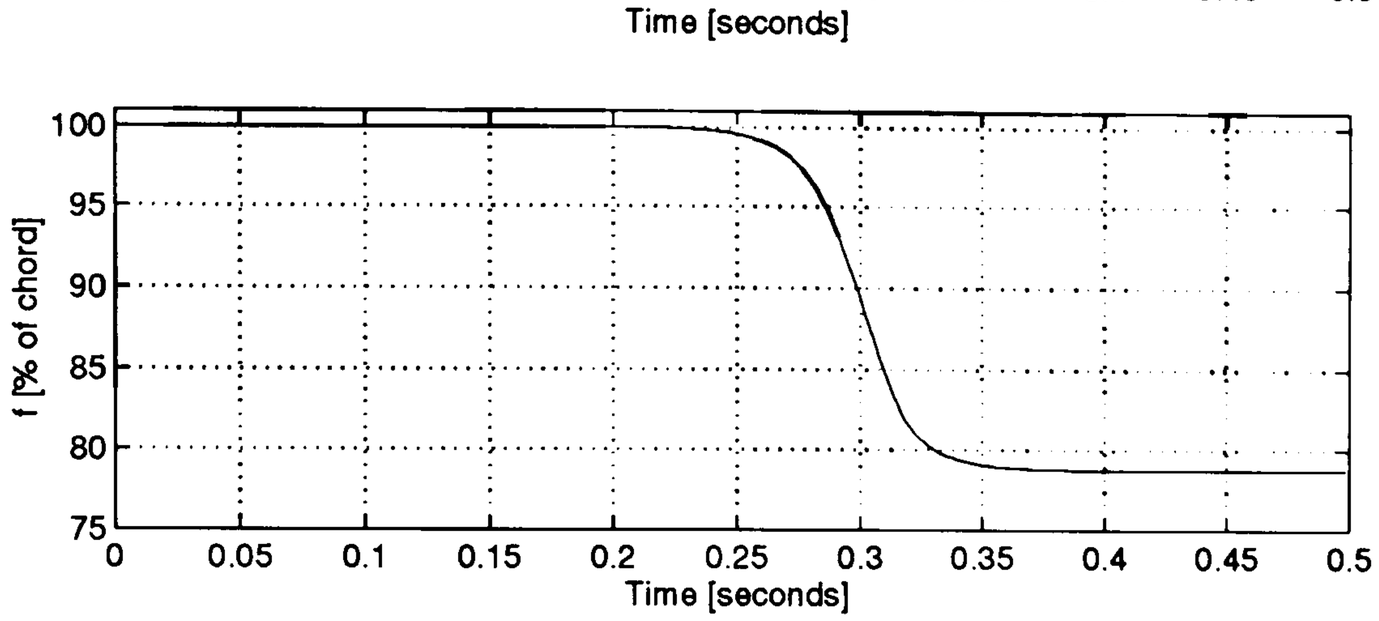


Figure A.17: Separation point excursion in relation to airfoil motion: Case in Figure 4.15 ($M = 0.55$).

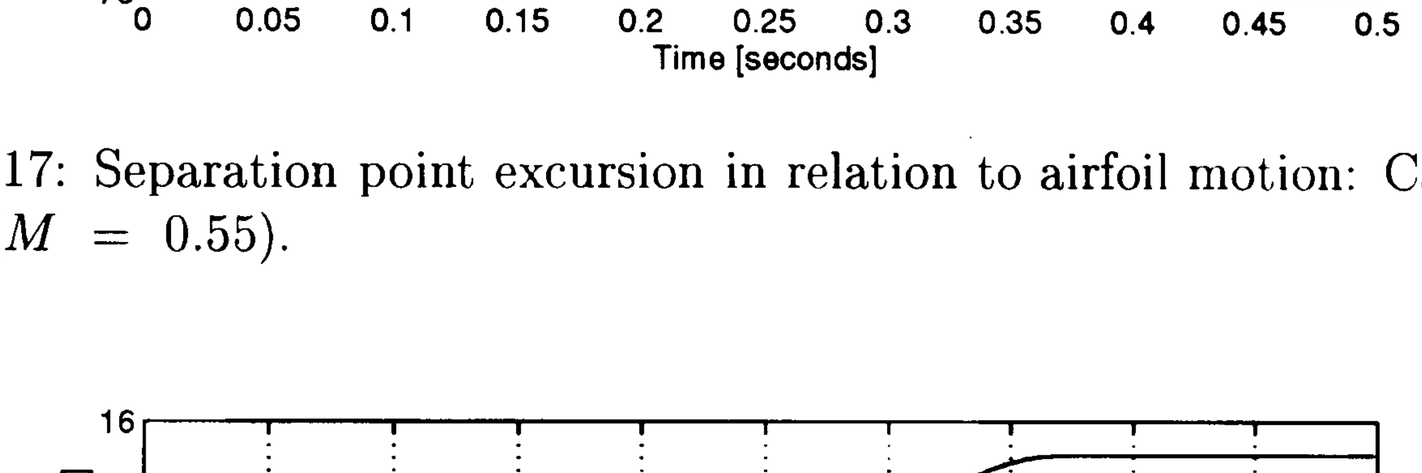


Figure A.18: Separation point excursion in relation to airfoil motion: Case in Figure 4.16 ($M = 0.25$).

Appendix B

Unsteady Transonic Pressure Distributions

The figures presented in this Appendix are obtained from a CFD code based on the Euler equations and developed by Dubuc *et al.* [1997], for unsteady aerodynamic response in the transonic regime of two-dimensional airfoils. Each case presents the pressure distribution for different points in the motion history. Figure B.1 illustrates the main parameters used to generate the cases.

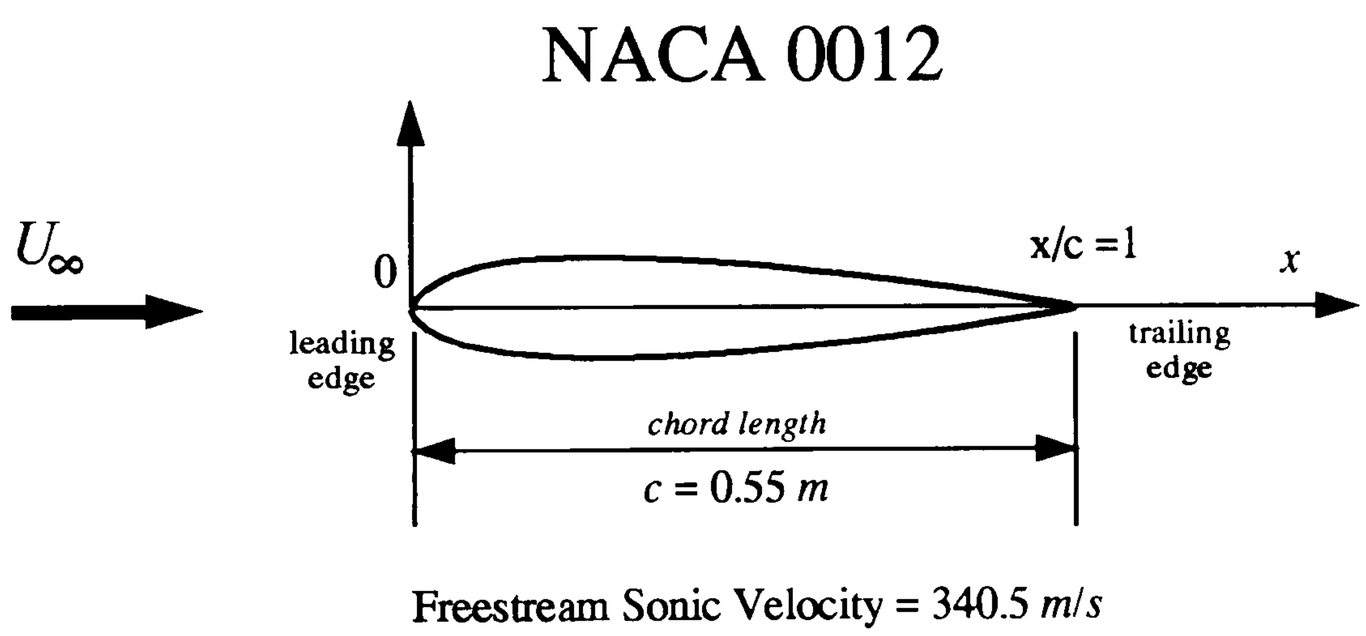
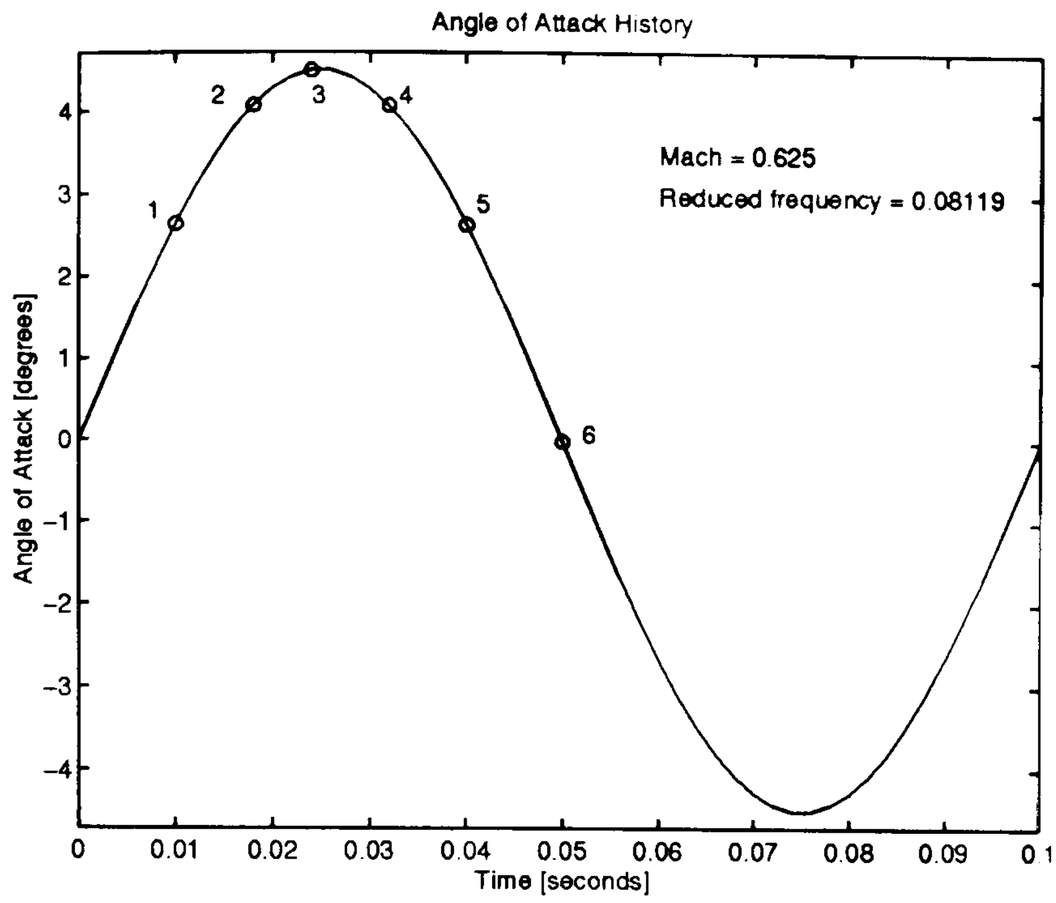


Figure B.1: Parameters for the calculation of pressure distributions for the unsteady transonic cases.



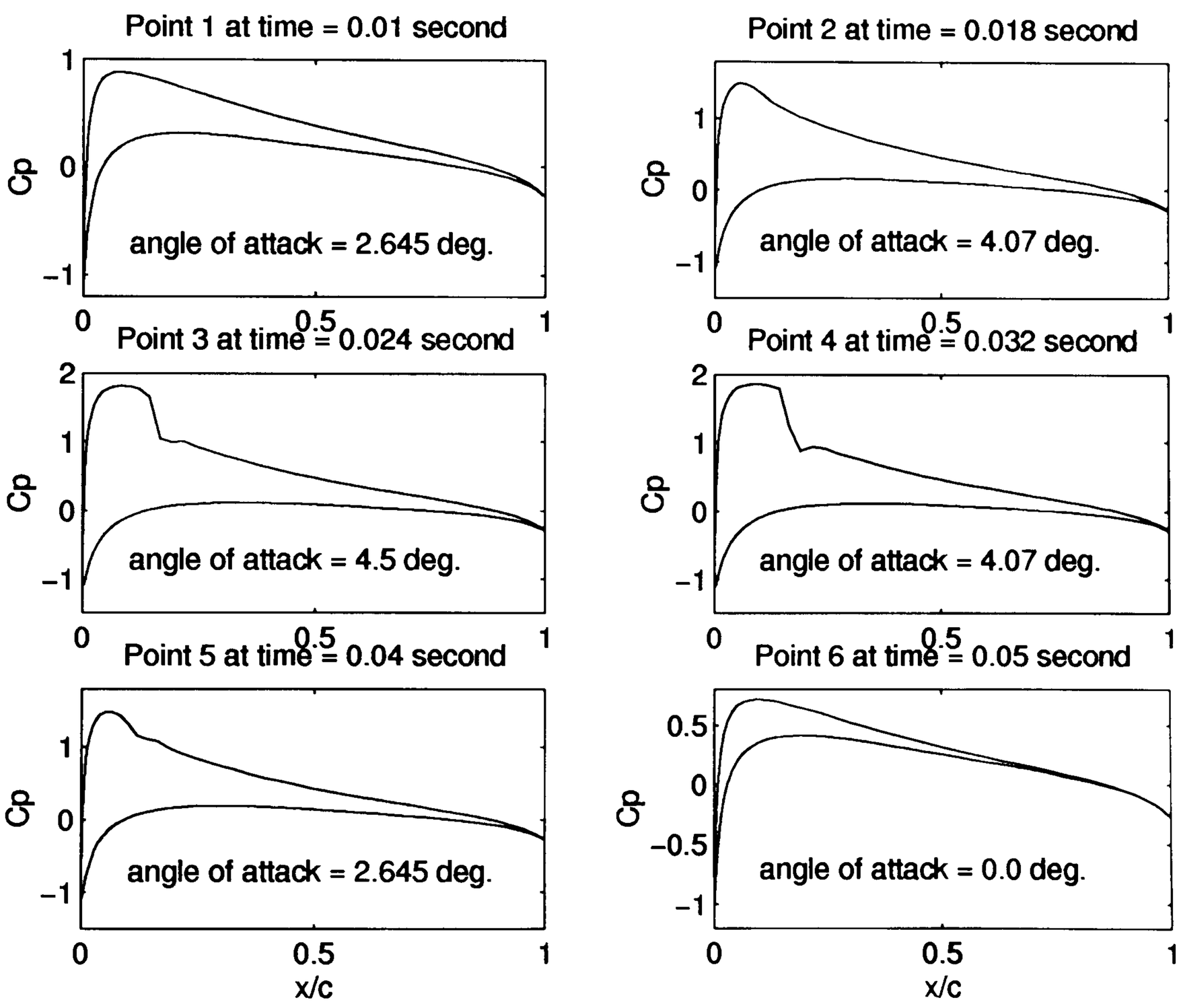
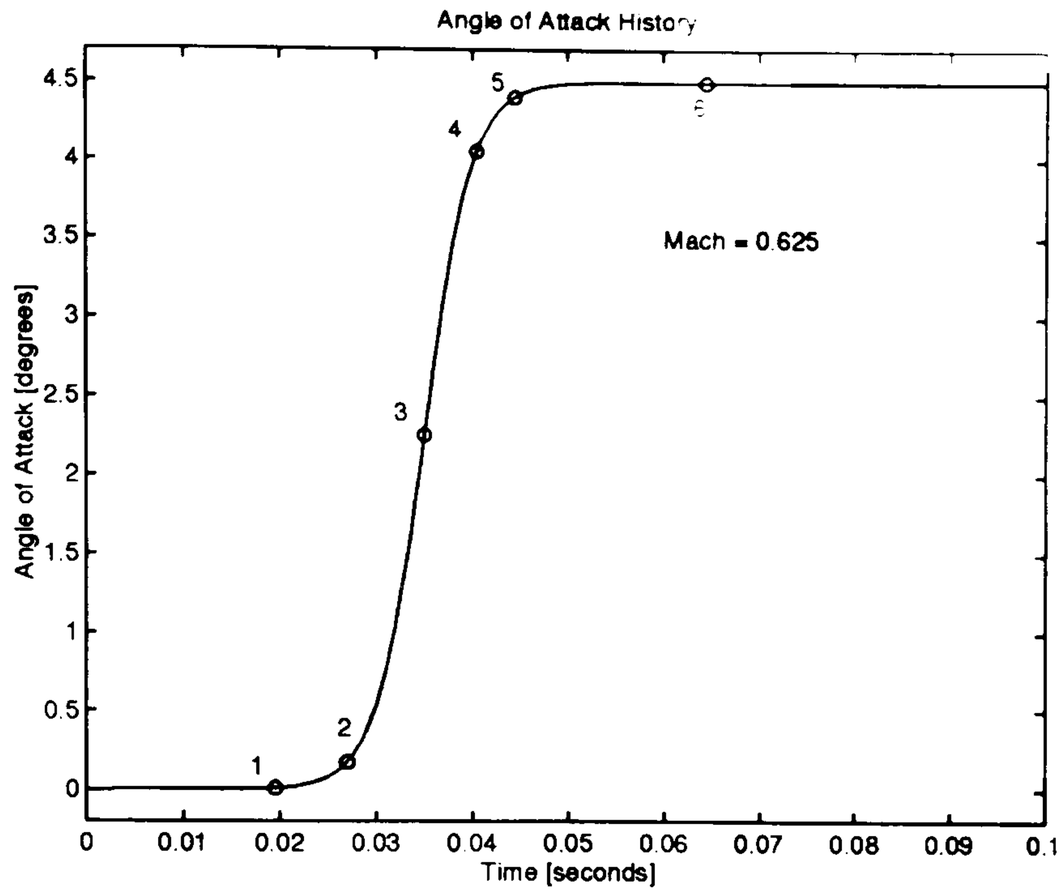


Figure B.2: Pressure distribution variation in relation to airfoil motion: Case in Figure 4.31 ($M = 0.625$).



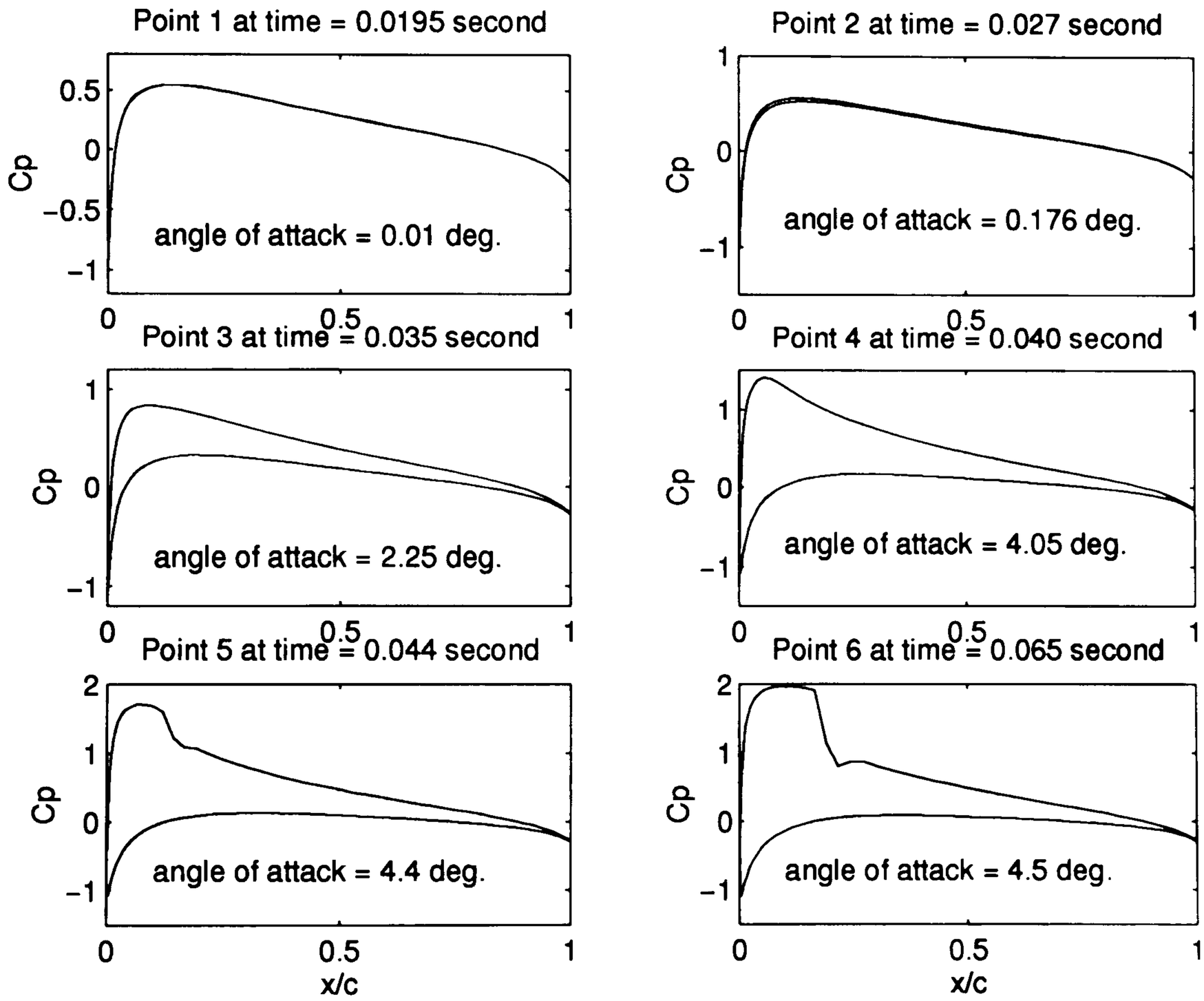
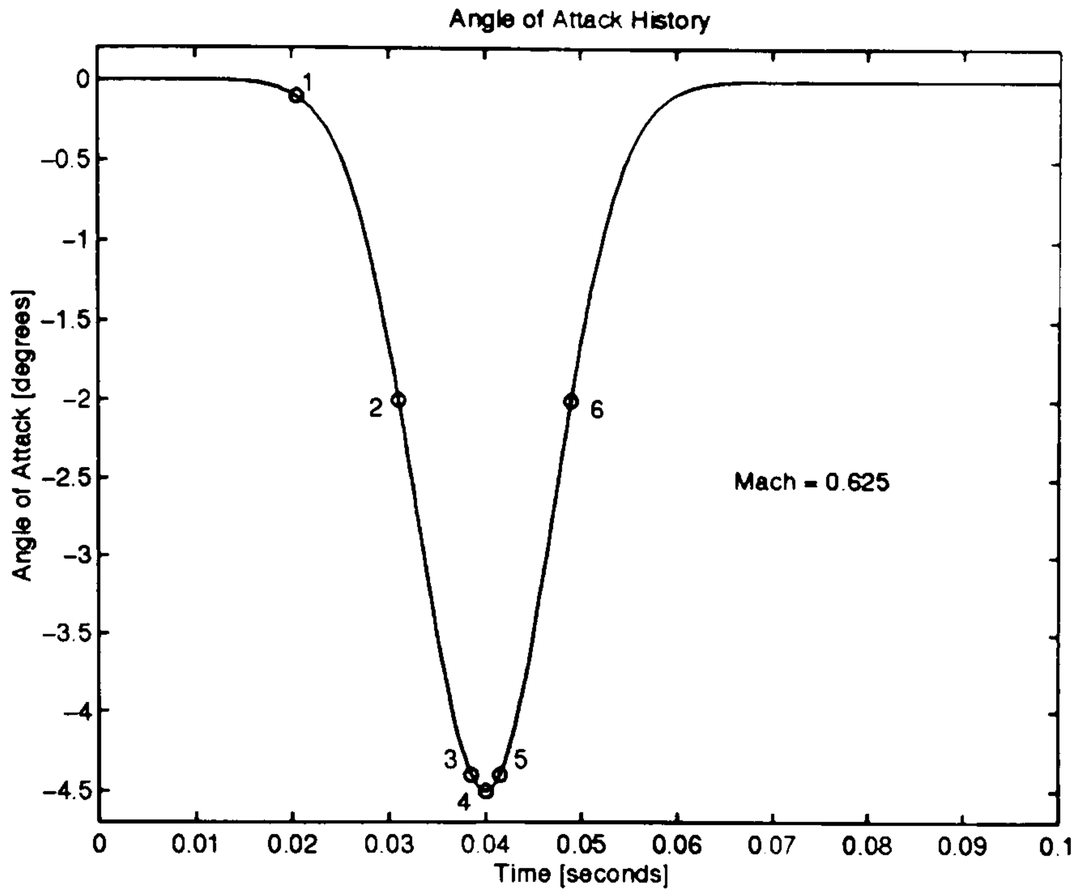


Figure B.3: Pressure distribution variation in relation to airfoil motion: Case in Figure 4.32 ($M = 0.625$).



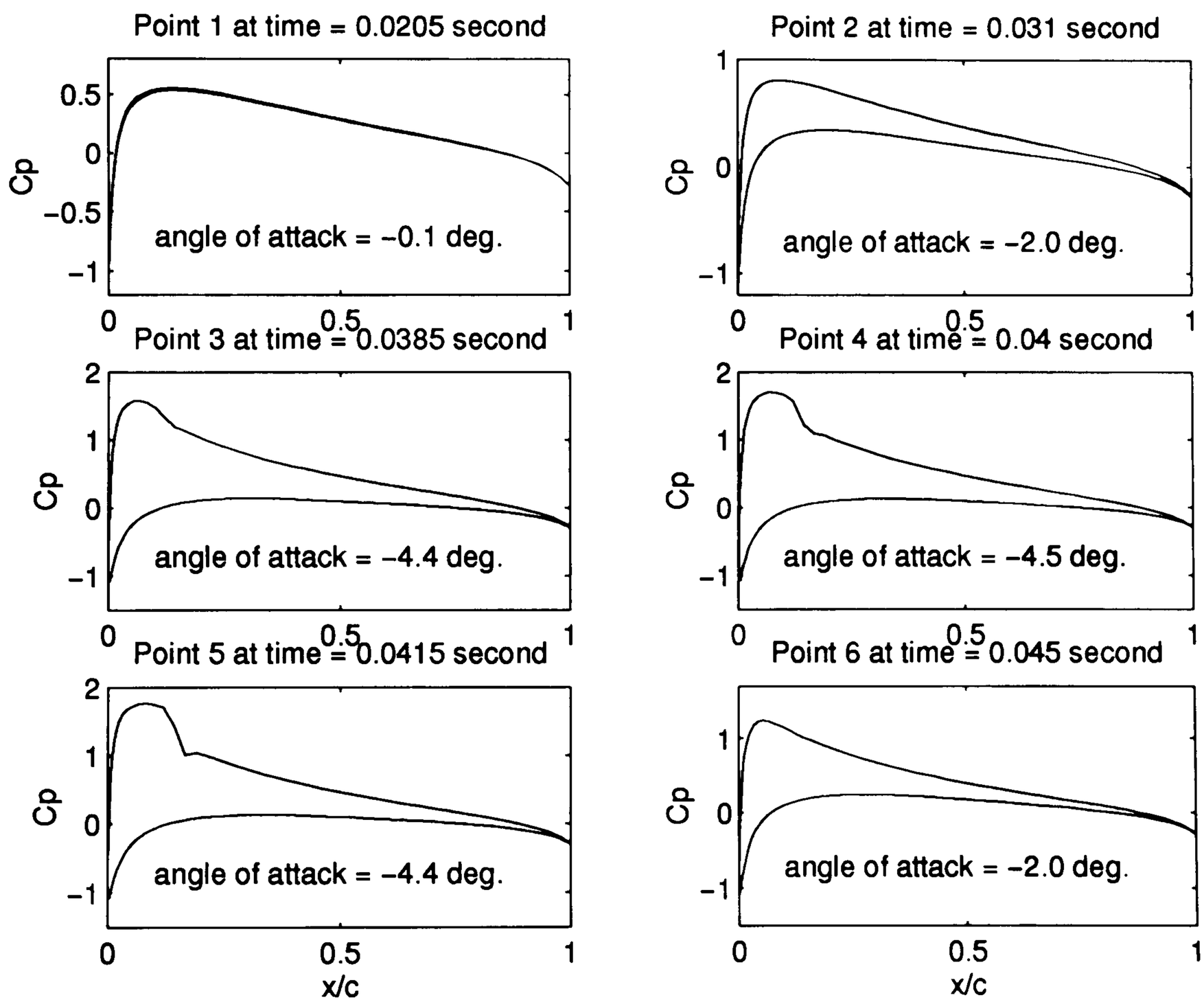
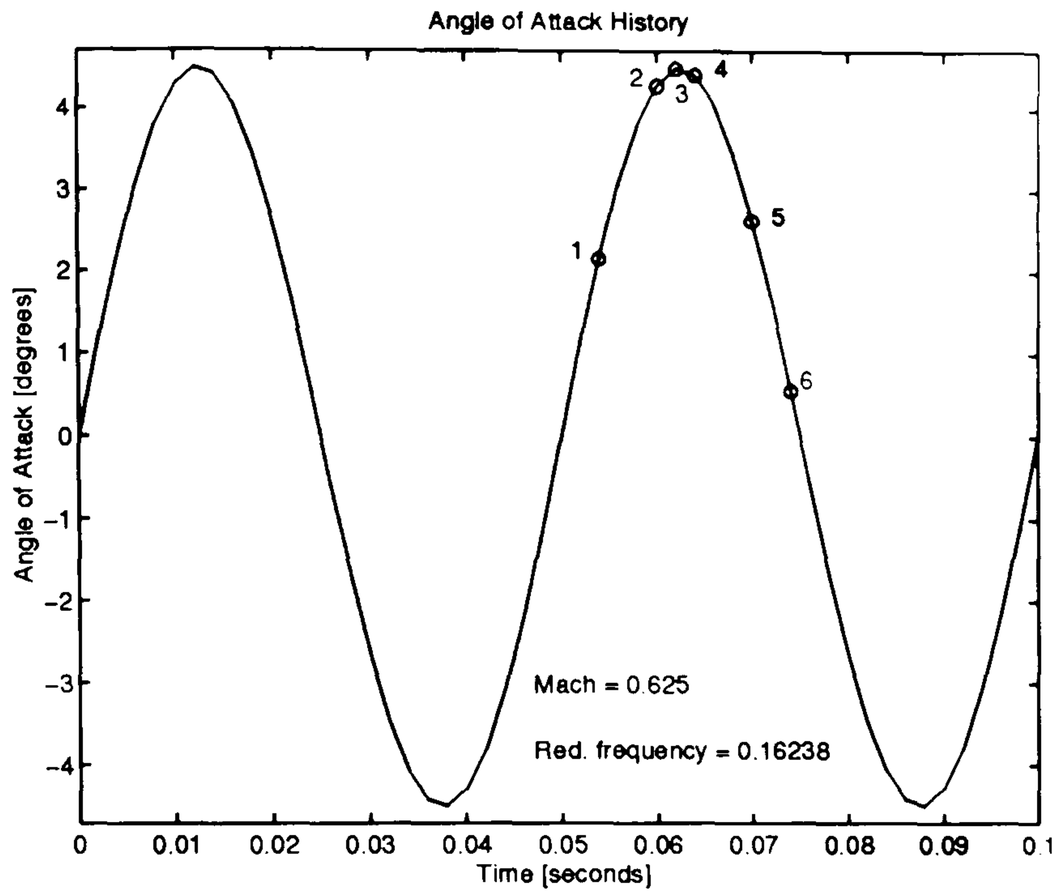


Figure B.4: Pressure distribution variation in relation to airfoil motion: Case in Figure 4.33 ($M = 0.625$).



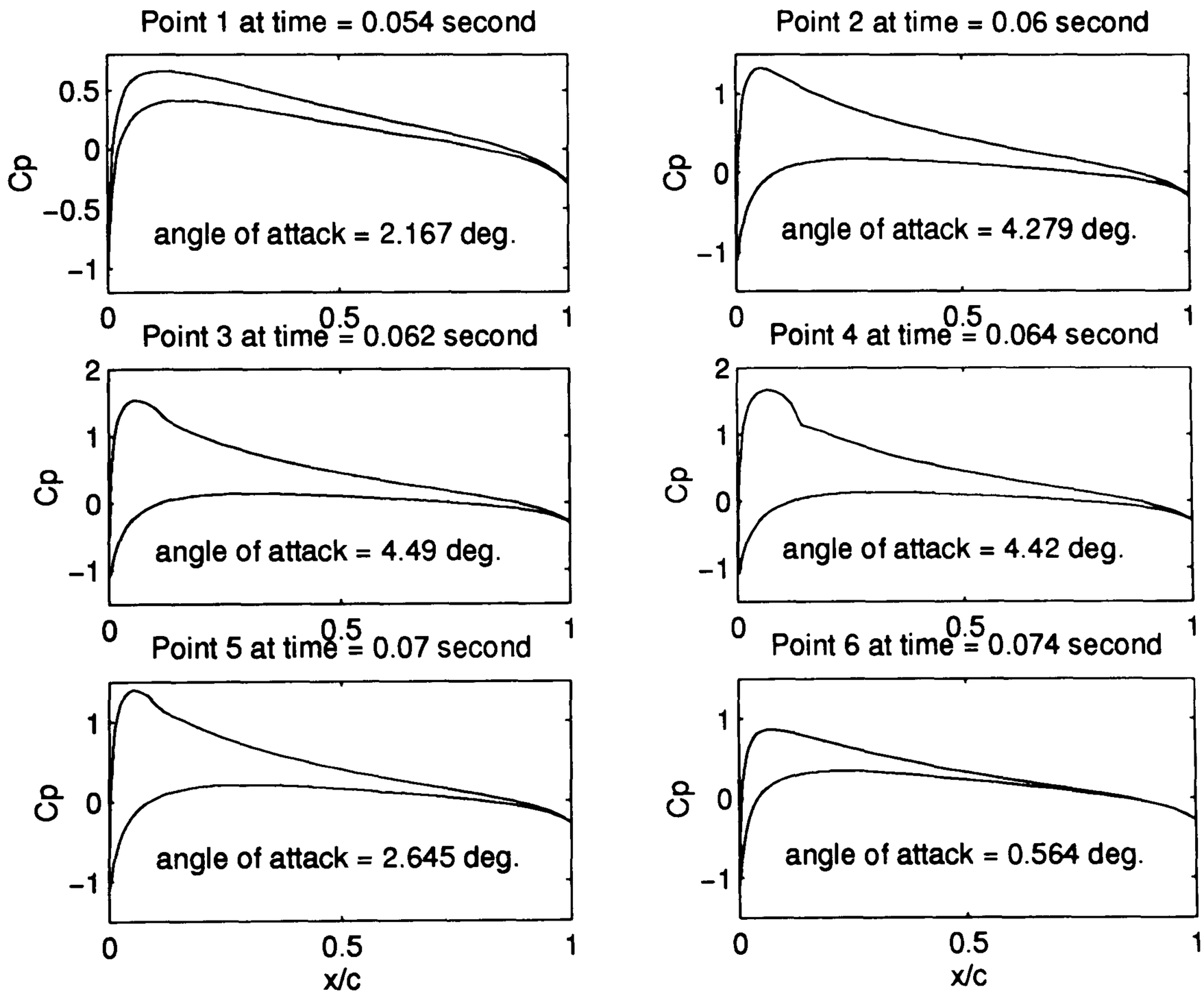
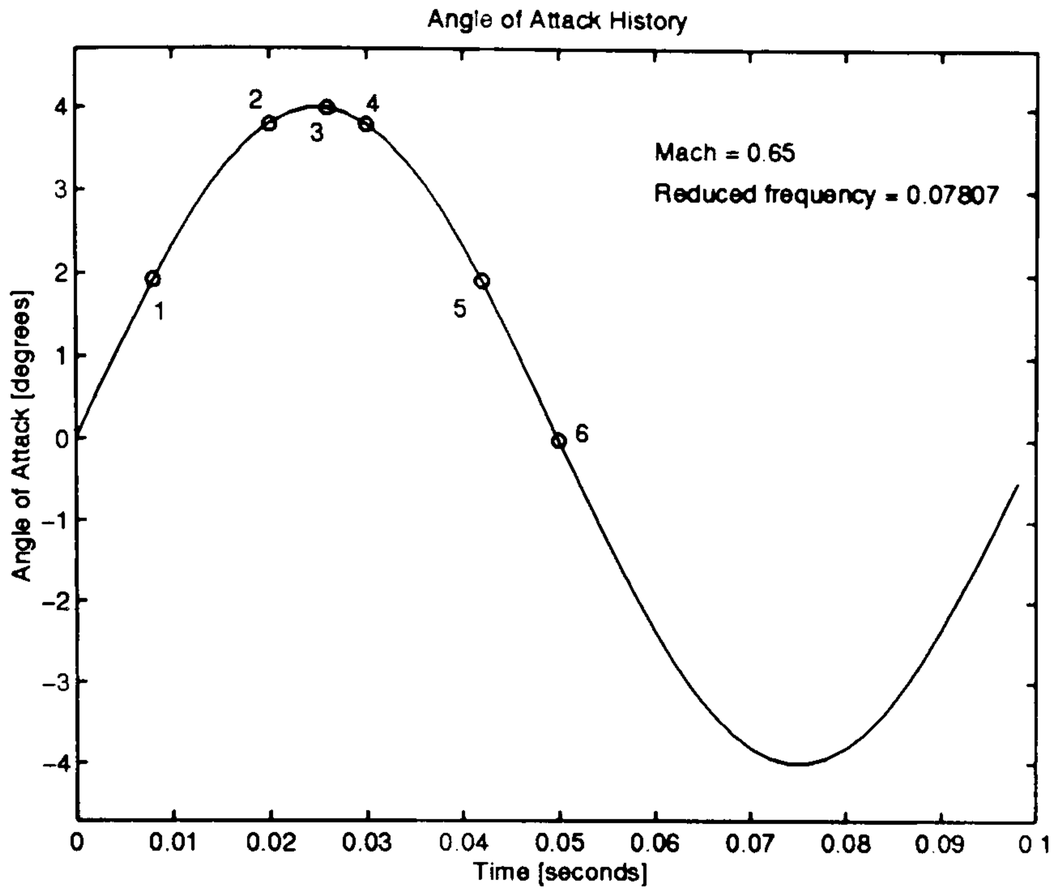


Figure B.5: Pressure distribution variation in relation to airfoil motion: Case in Figure 4.43 ($M = 0.625$).



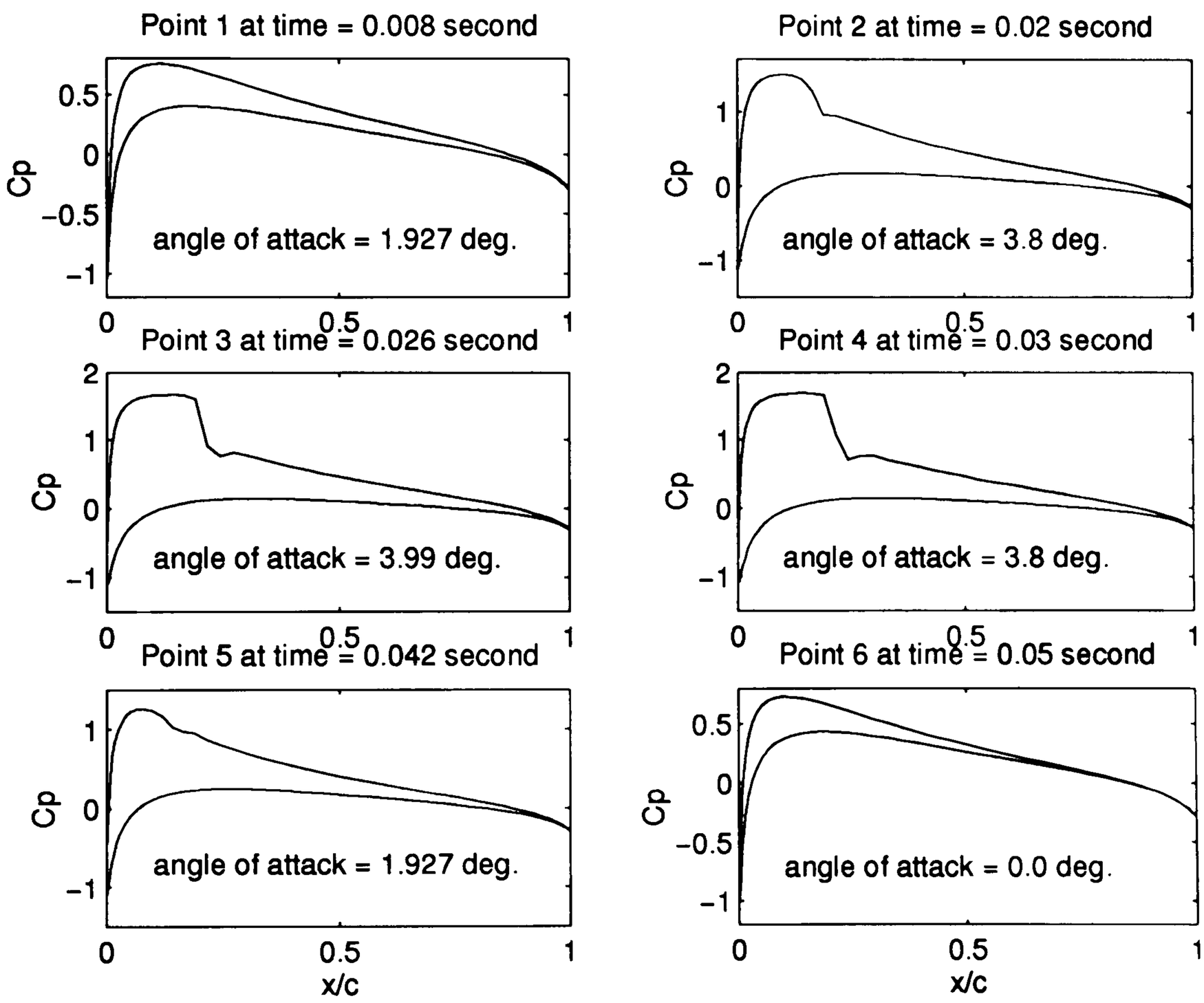
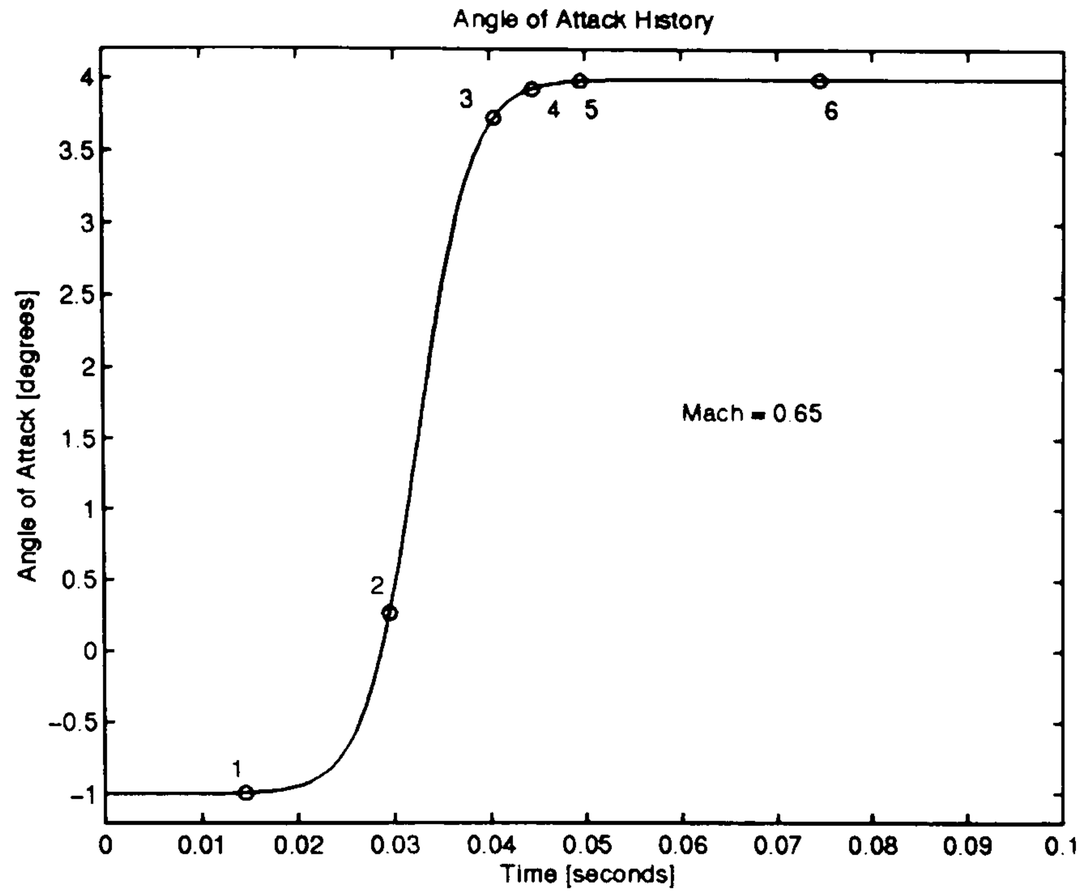


Figure B.6: Pressure distribution variation in relation to airfoil motion: Case in Figure 4.18 ($M = 0.65$).



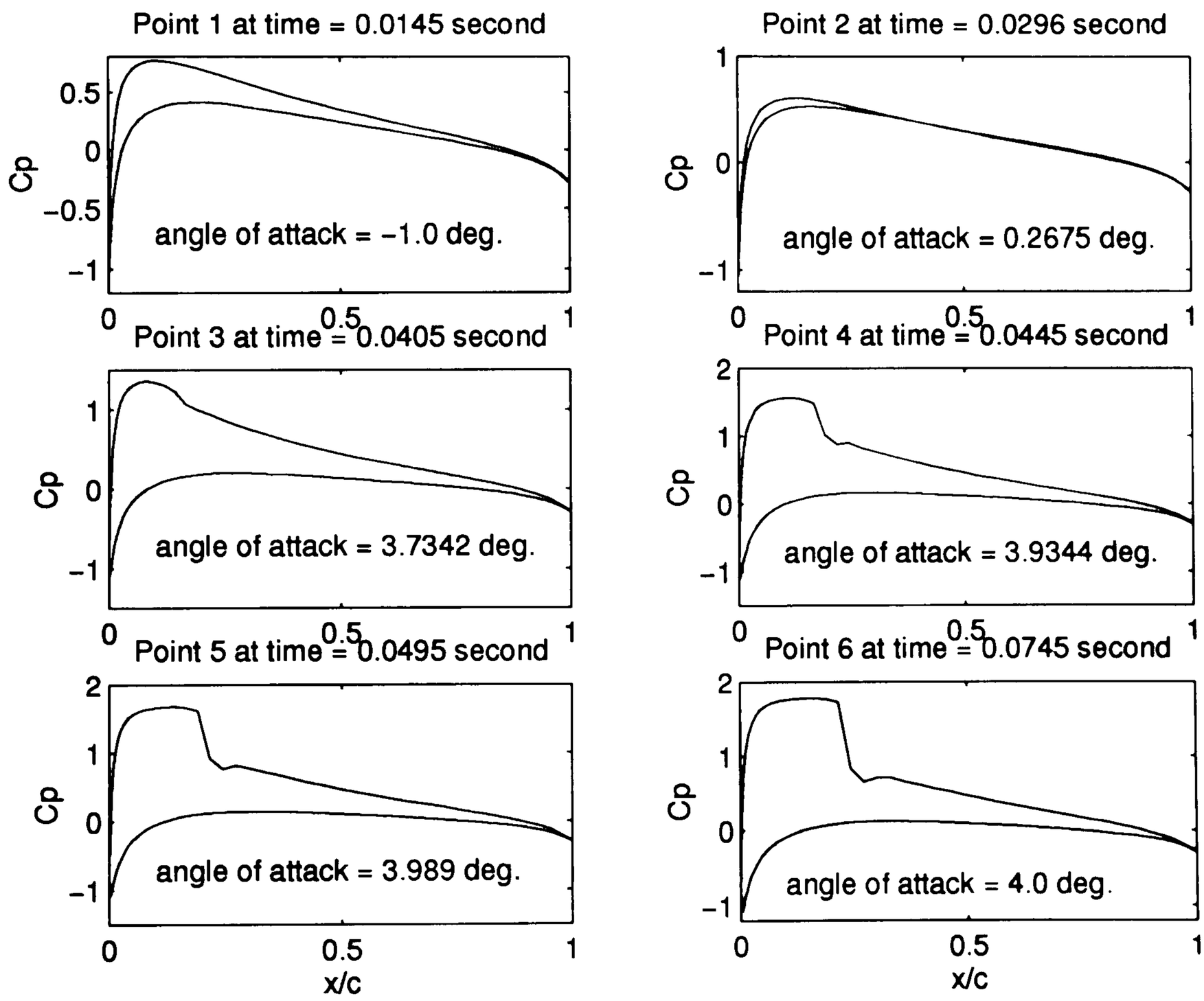
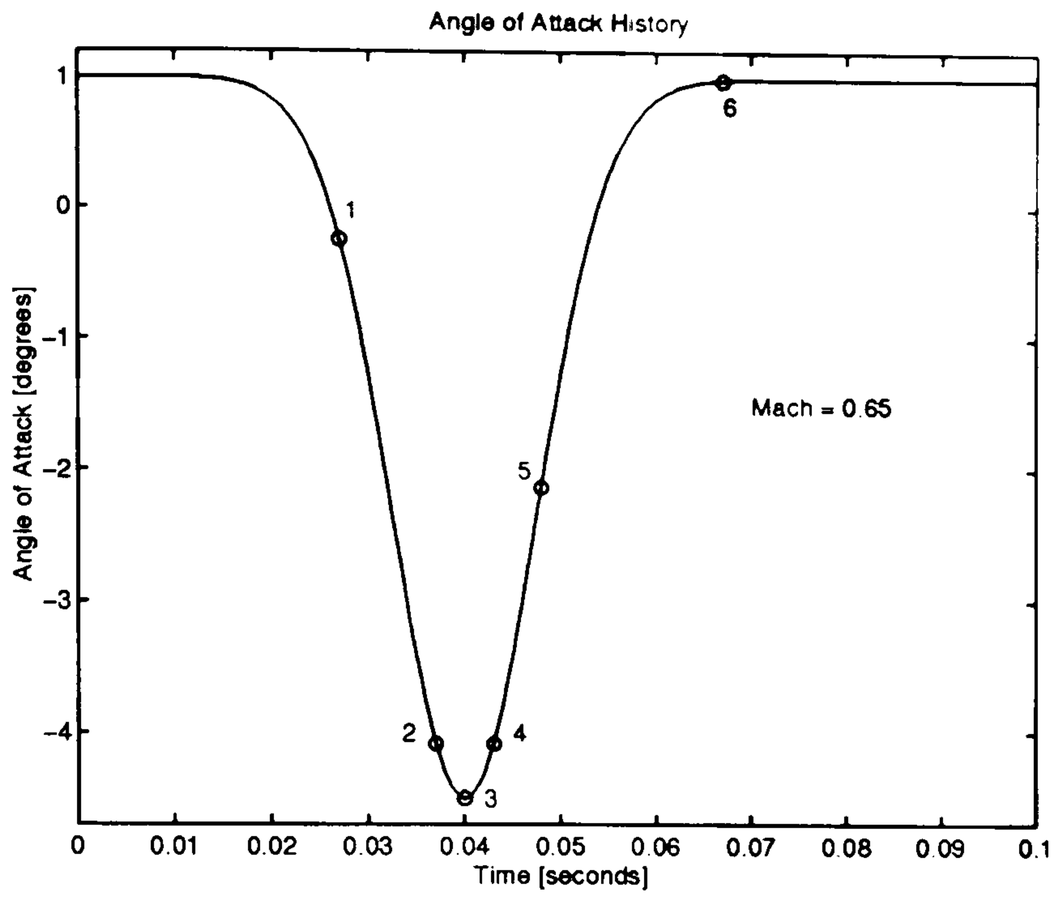


Figure B.7: Pressure distribution variation in relation to airfoil motion: Case in Figure 4.19 ($M = 0.65$).



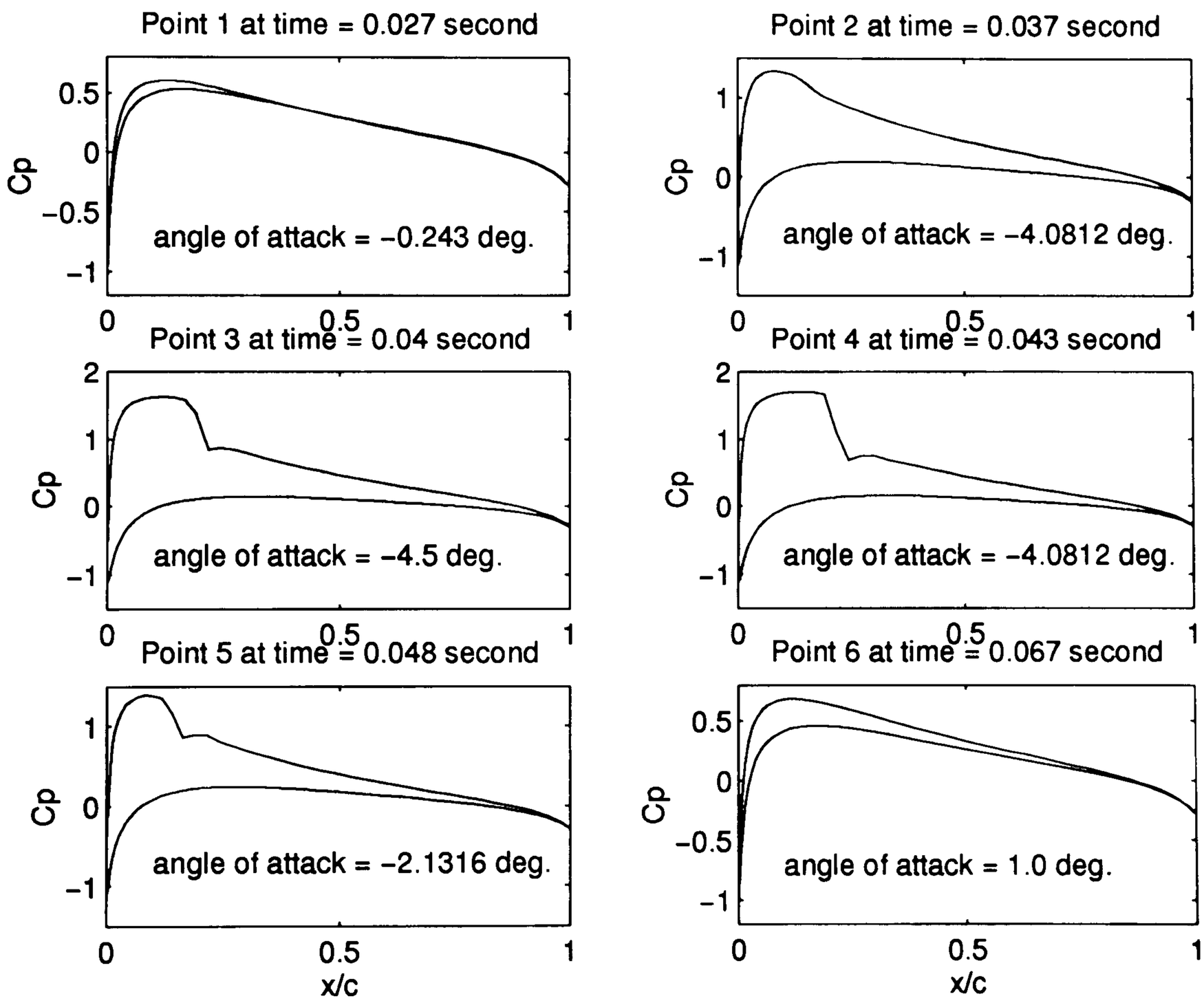
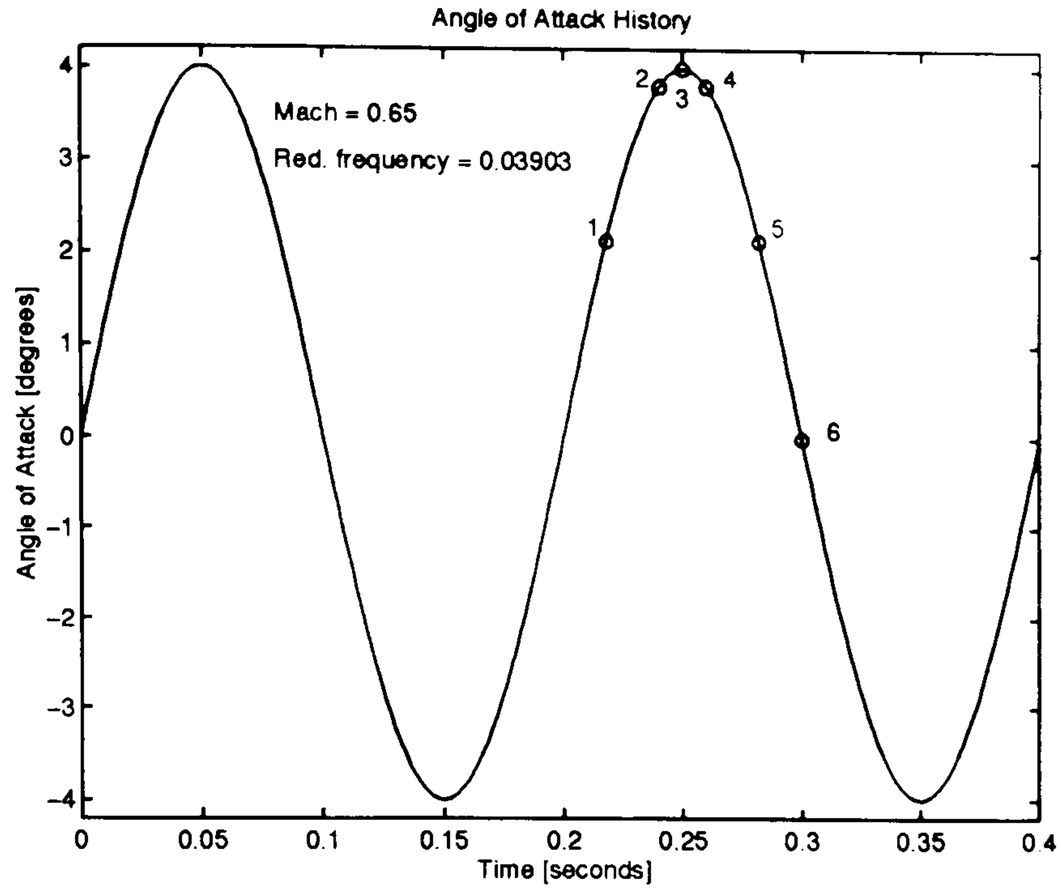


Figure B.8: Pressure distribution variation in relation to airfoil motion: Case in Figure 4.20 ($M = 0.65$).



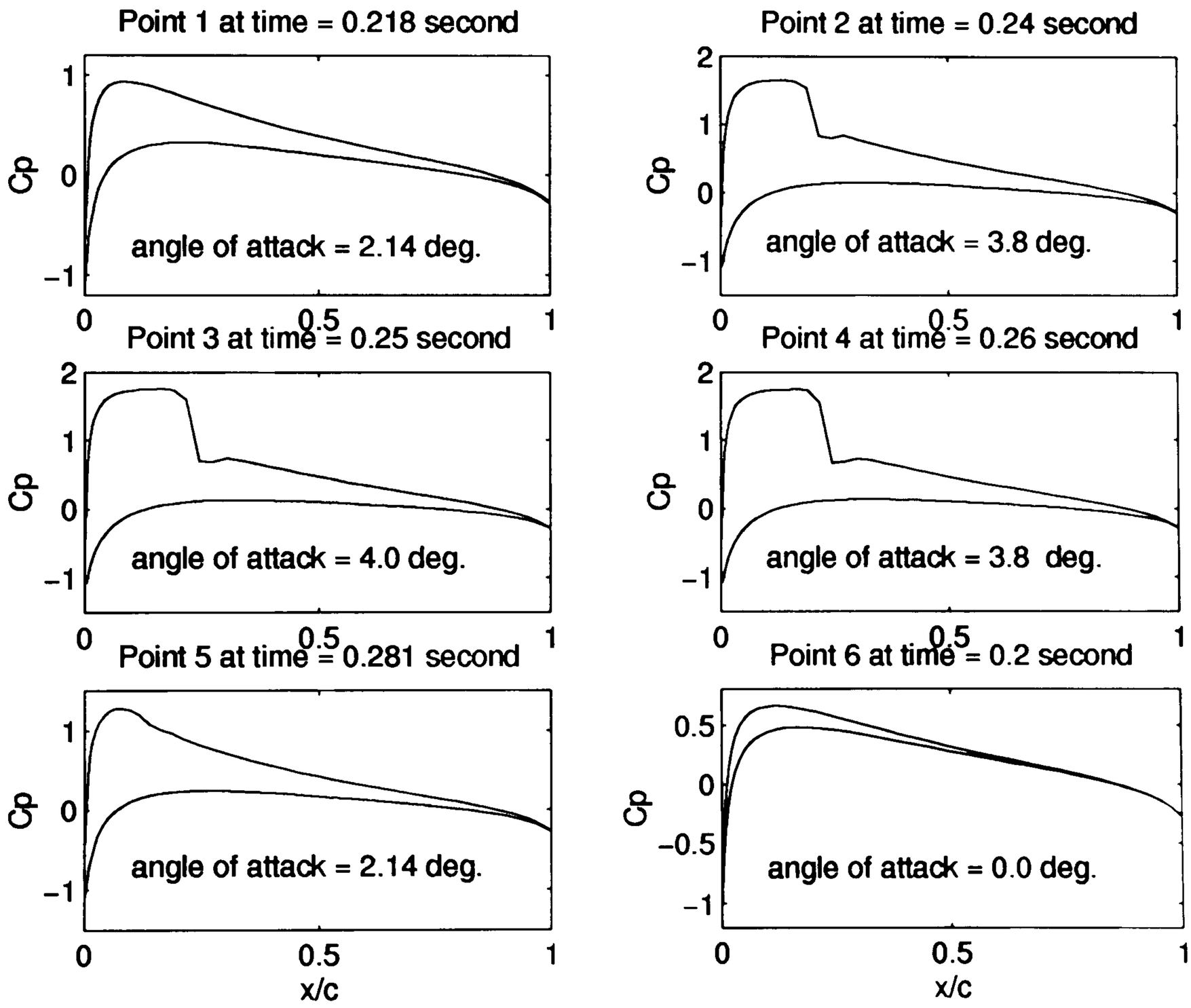
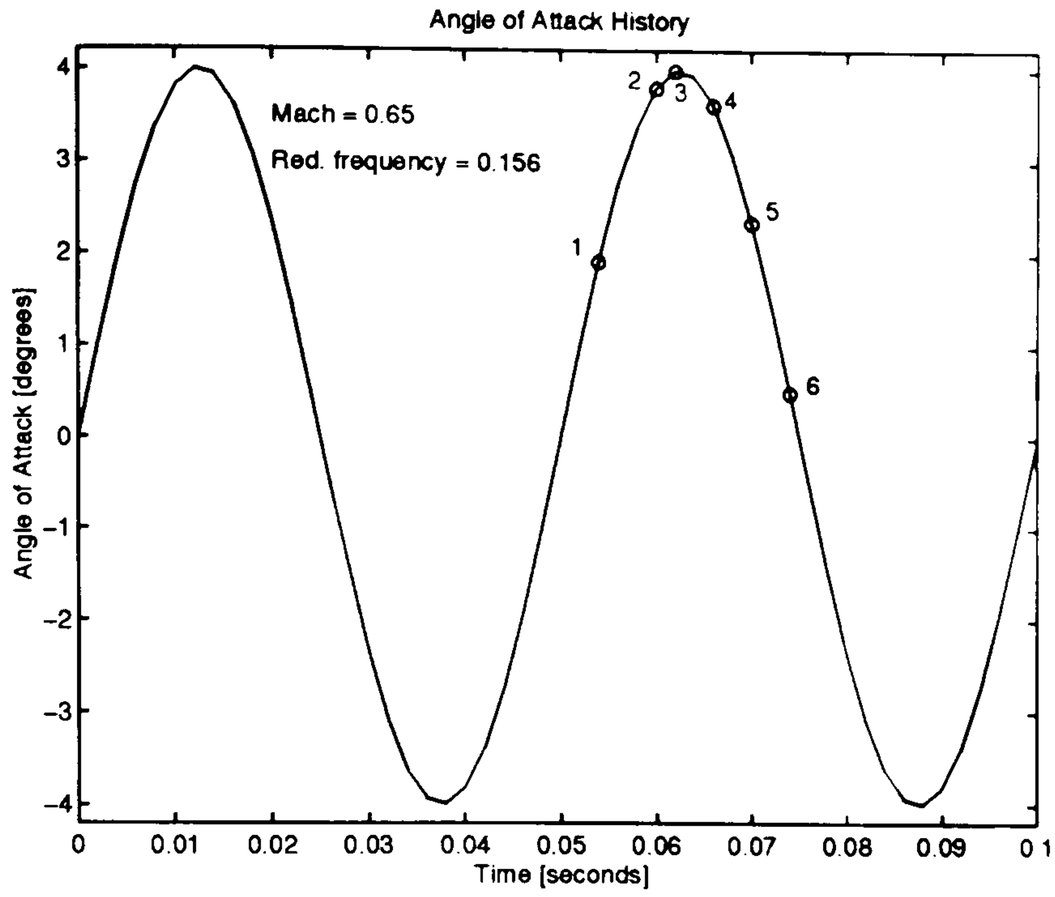


Figure B.9: Pressure distribution variation in relation to airfoil motion: Case in Figure 4.24 ($M = 0.65$).



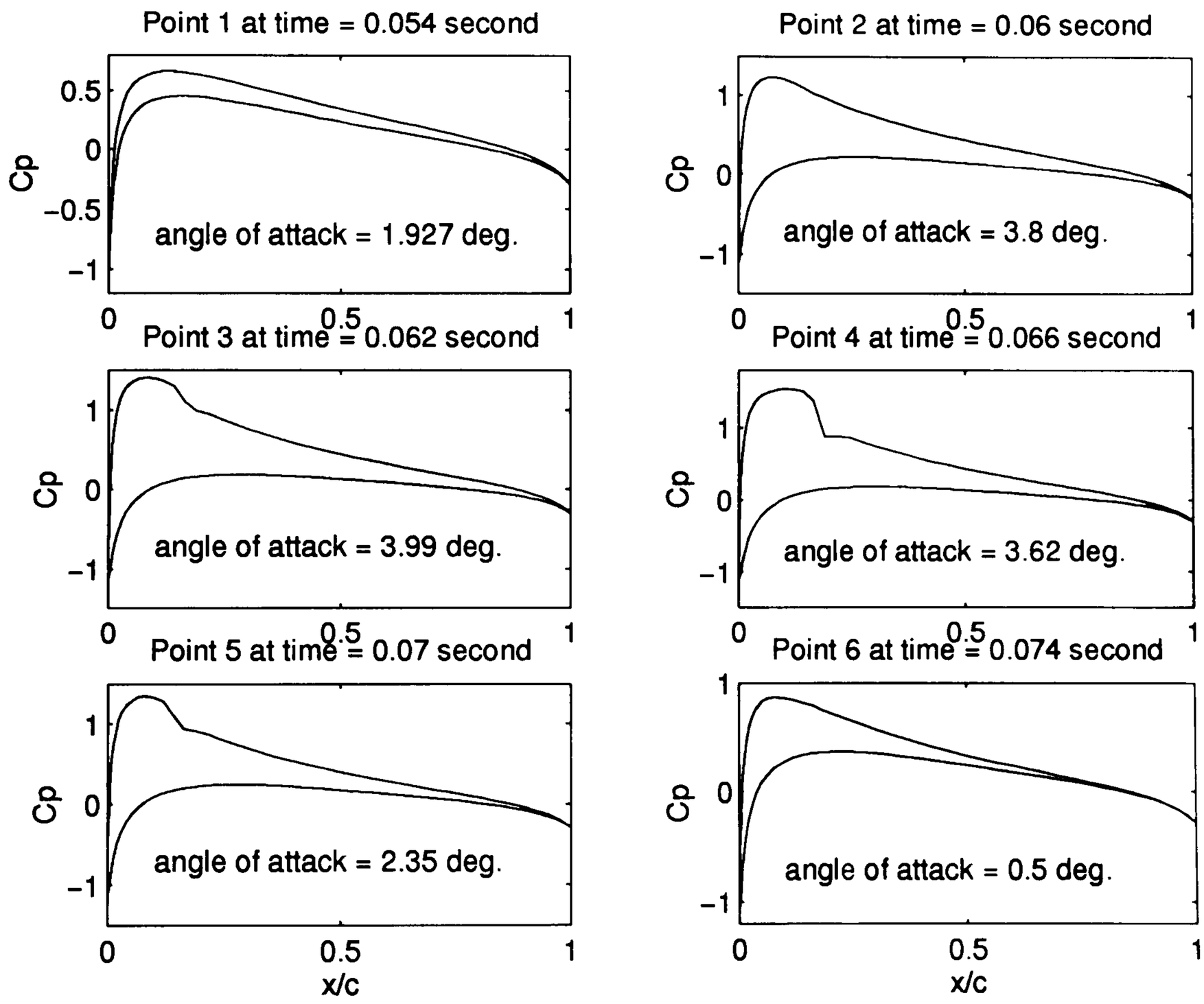
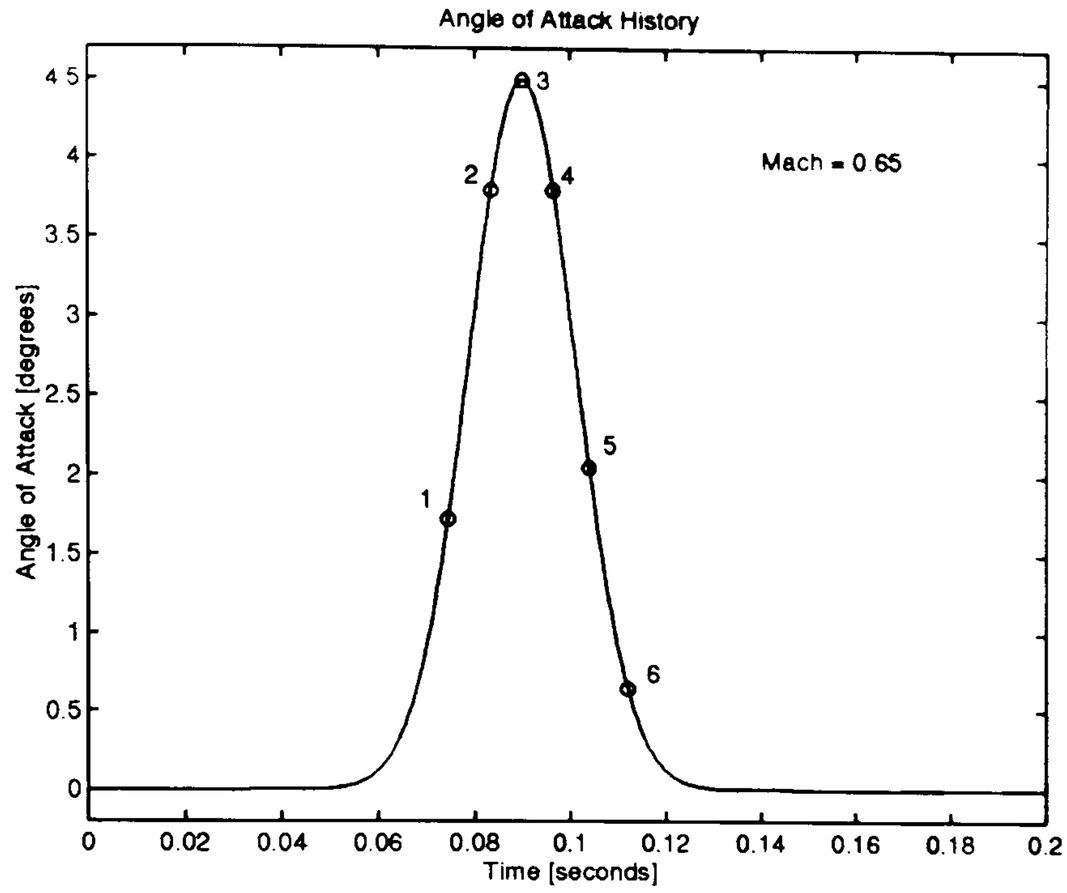


Figure B.10: Pressure distribution variation in relation to airfoil motion: Case in Figure 4.25 ($M = 0.65$).



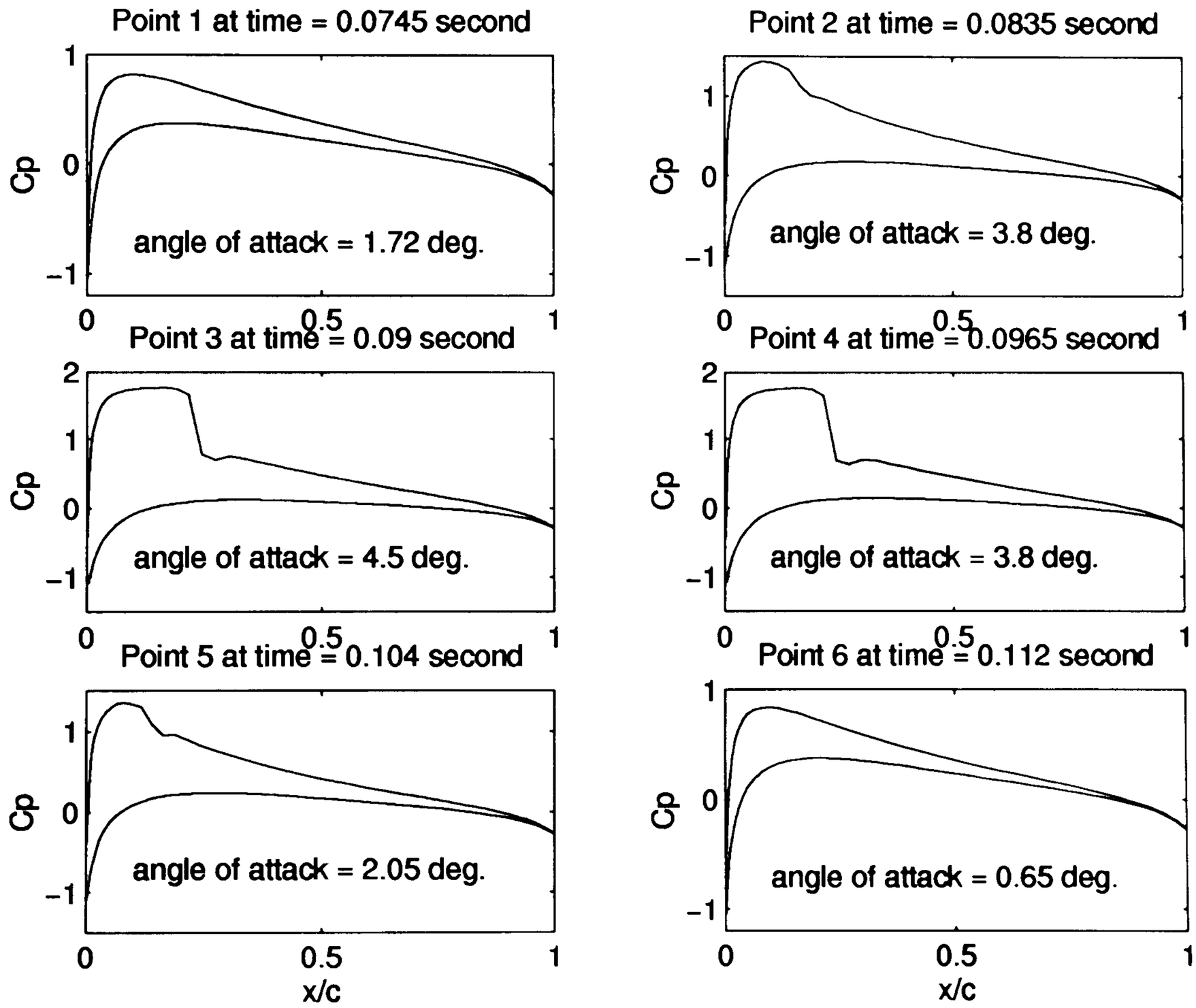
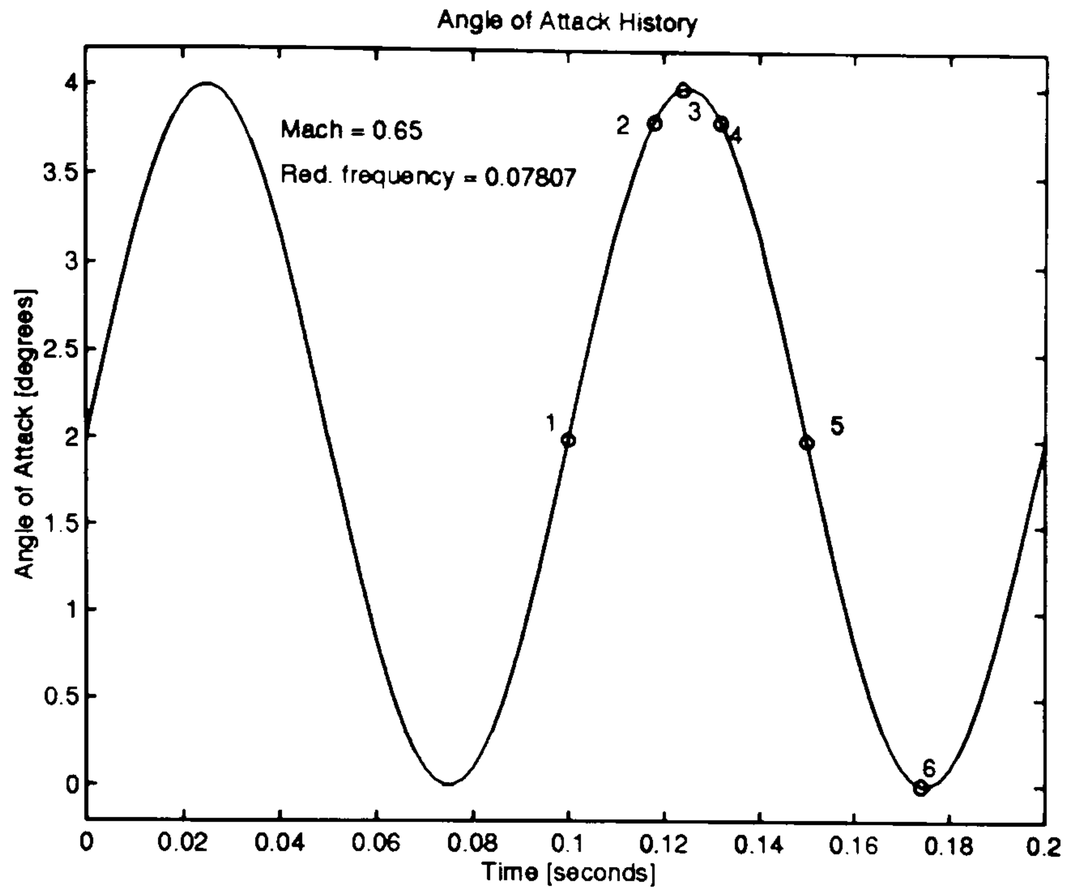


Figure B.11: Pressure distribution variation in relation to airfoil motion: Case in Figures 4.26 and 4.45 ($M = 0.65$).



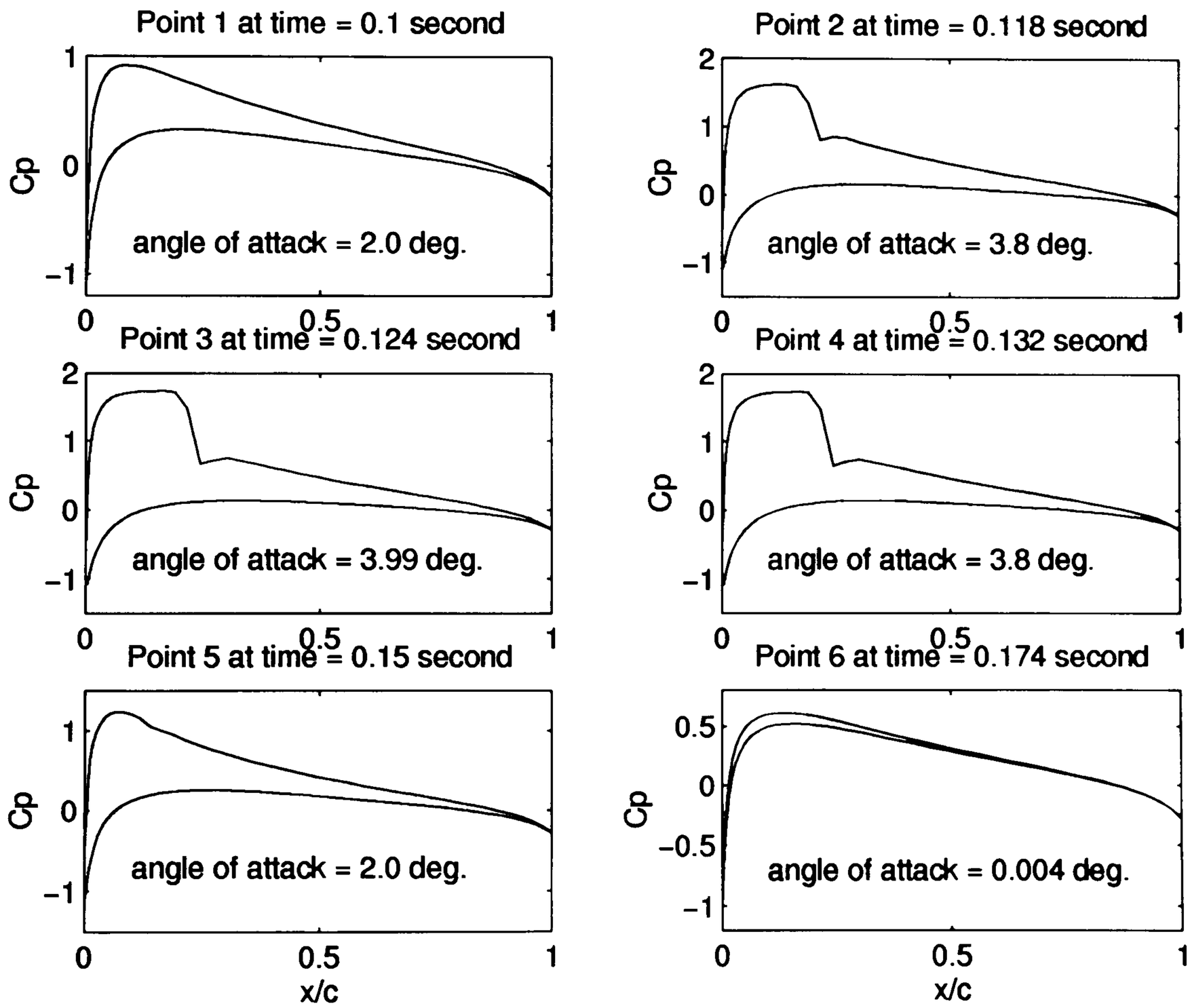
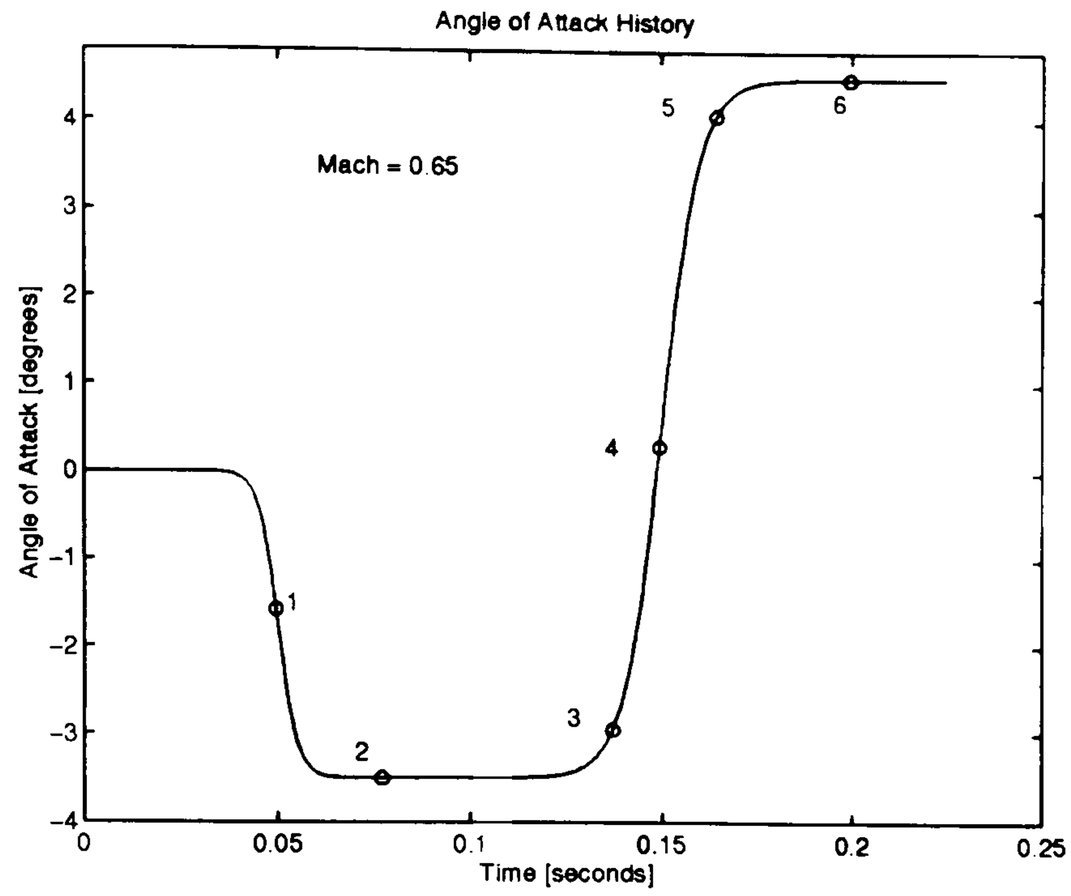


Figure B.12: Pressure distribution variation in relation to airfoil motion: Case in Figures 4.27 and 4.46 ($M = 0.65$).



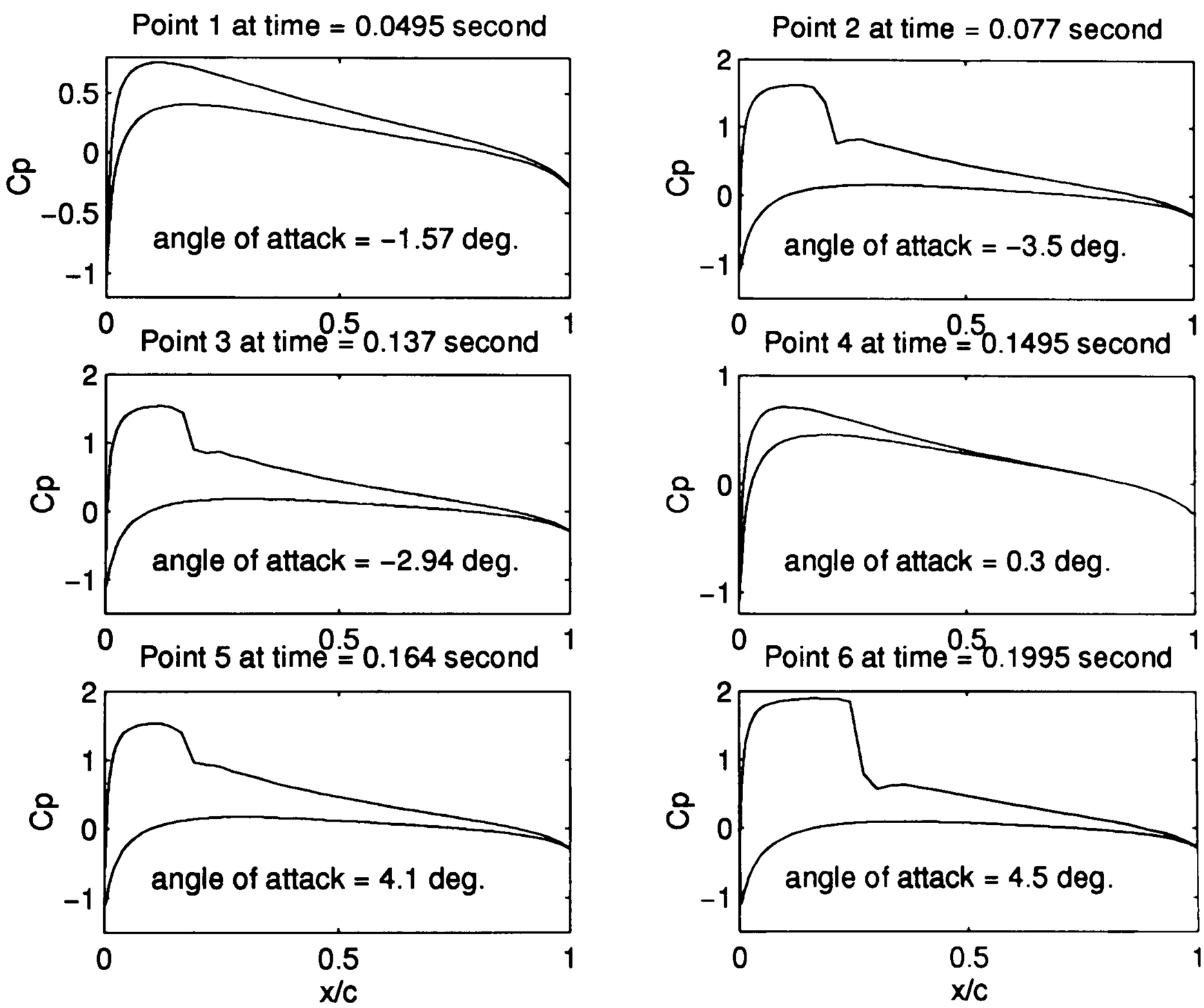
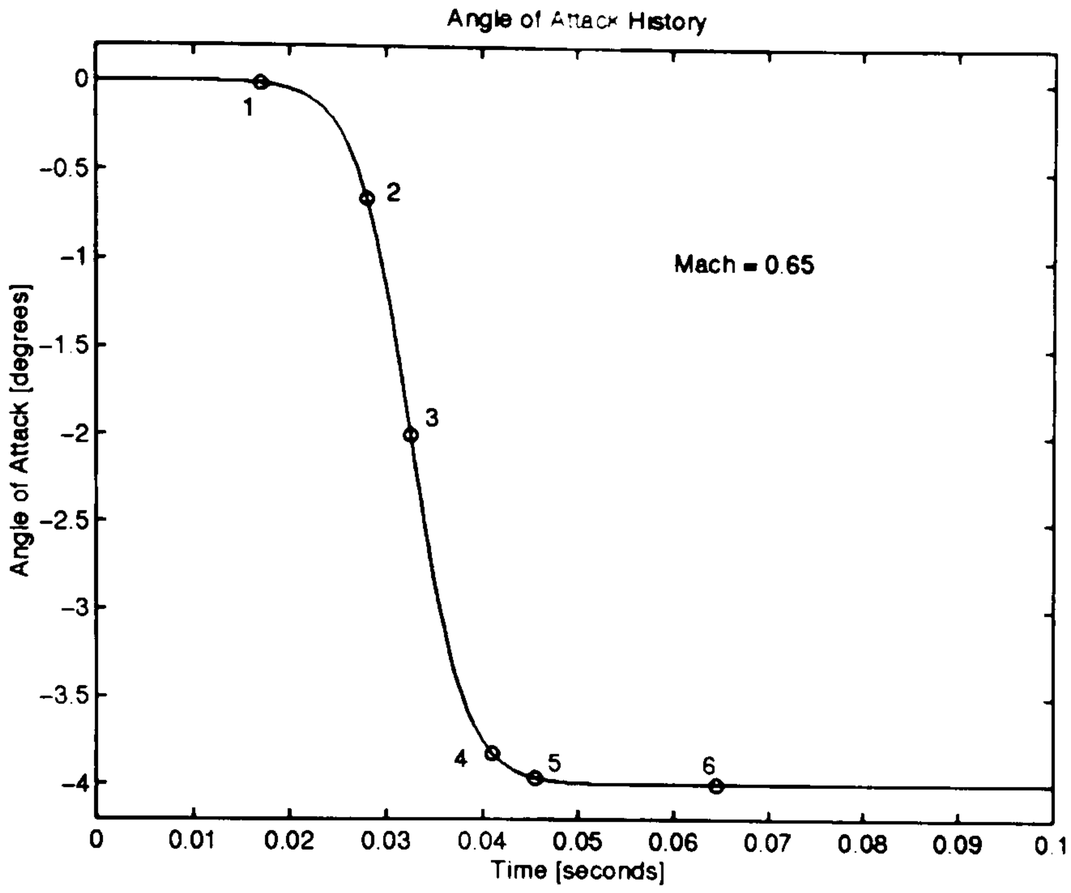


Figure B.13: Pressure distribution variation in relation to airfoil motion: Case in Figures 4.28 and 4.47 ($M = 0.65$).



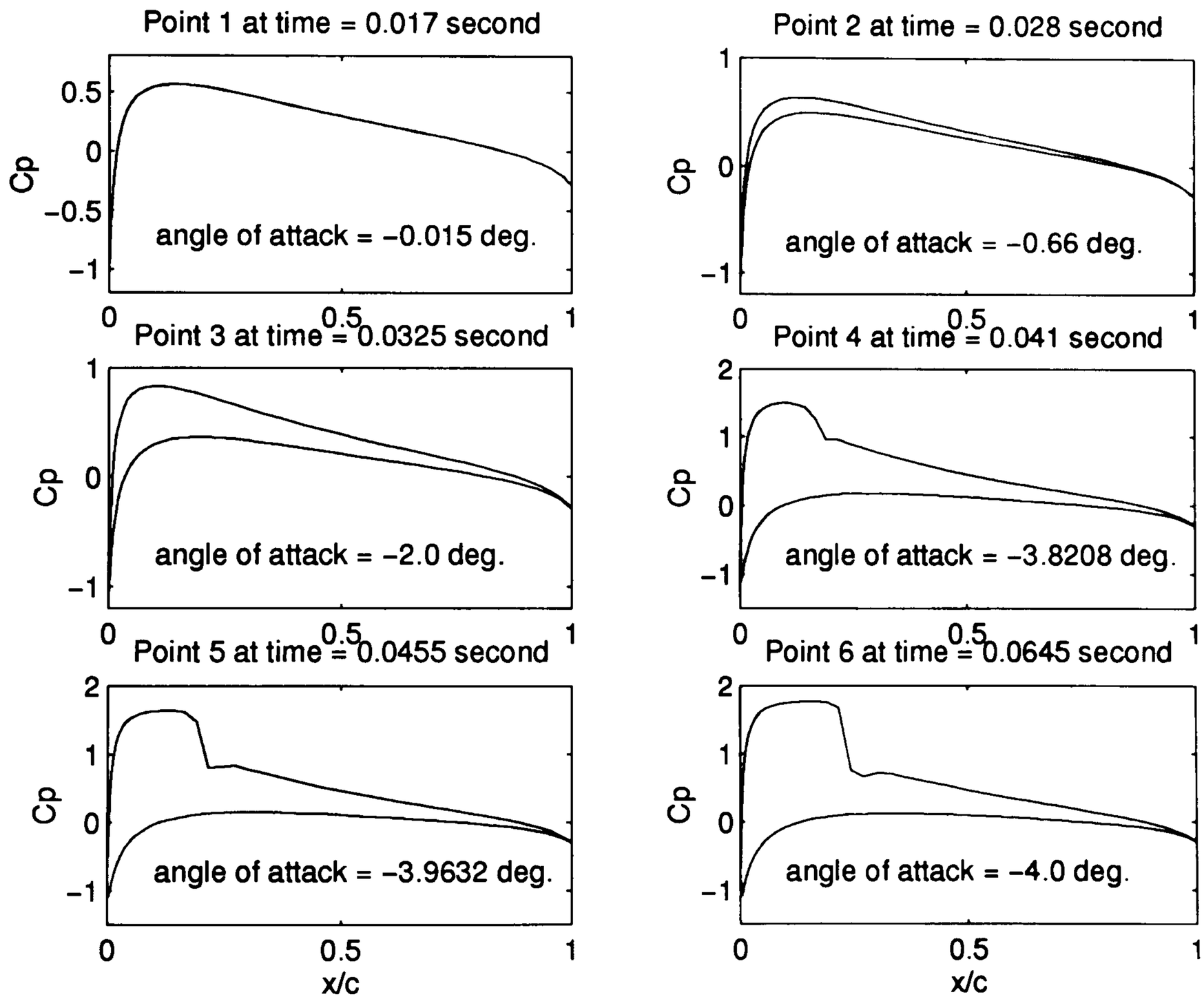
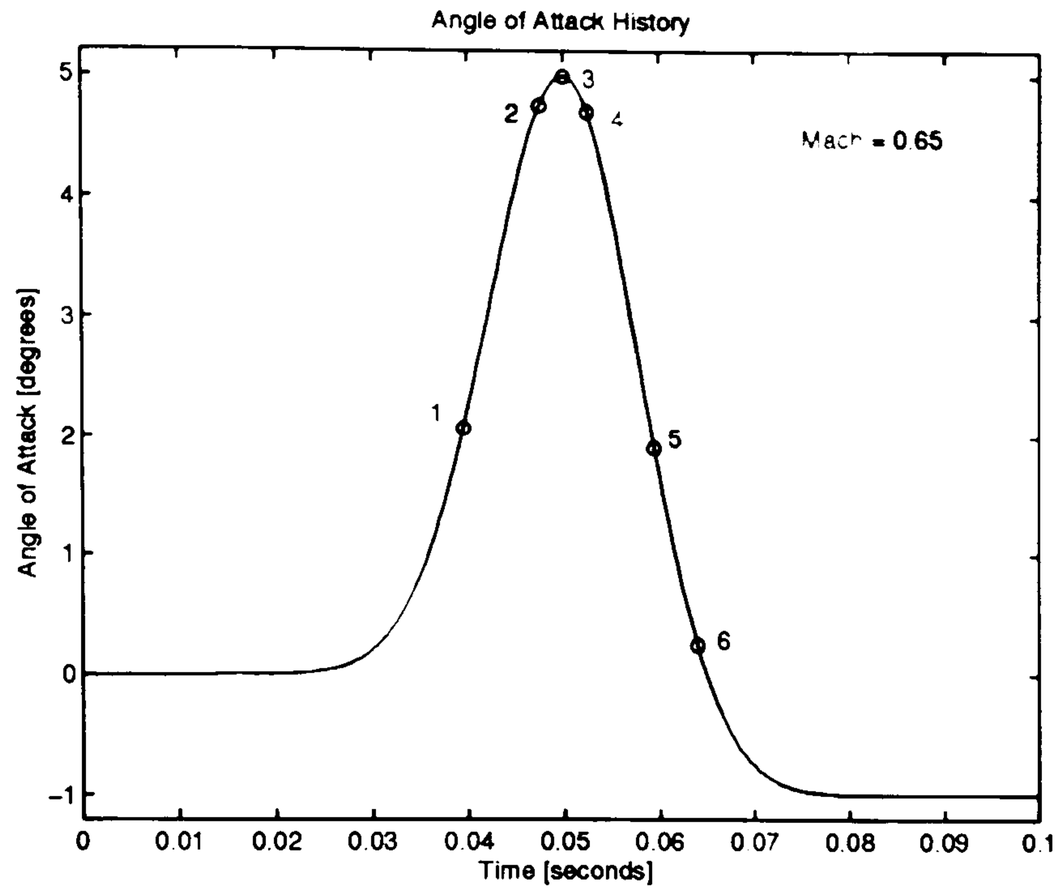


Figure B.14: Pressure distribution variation in relation to airfoil motion: Case in Figure 4.29 ($M = 0.65$).



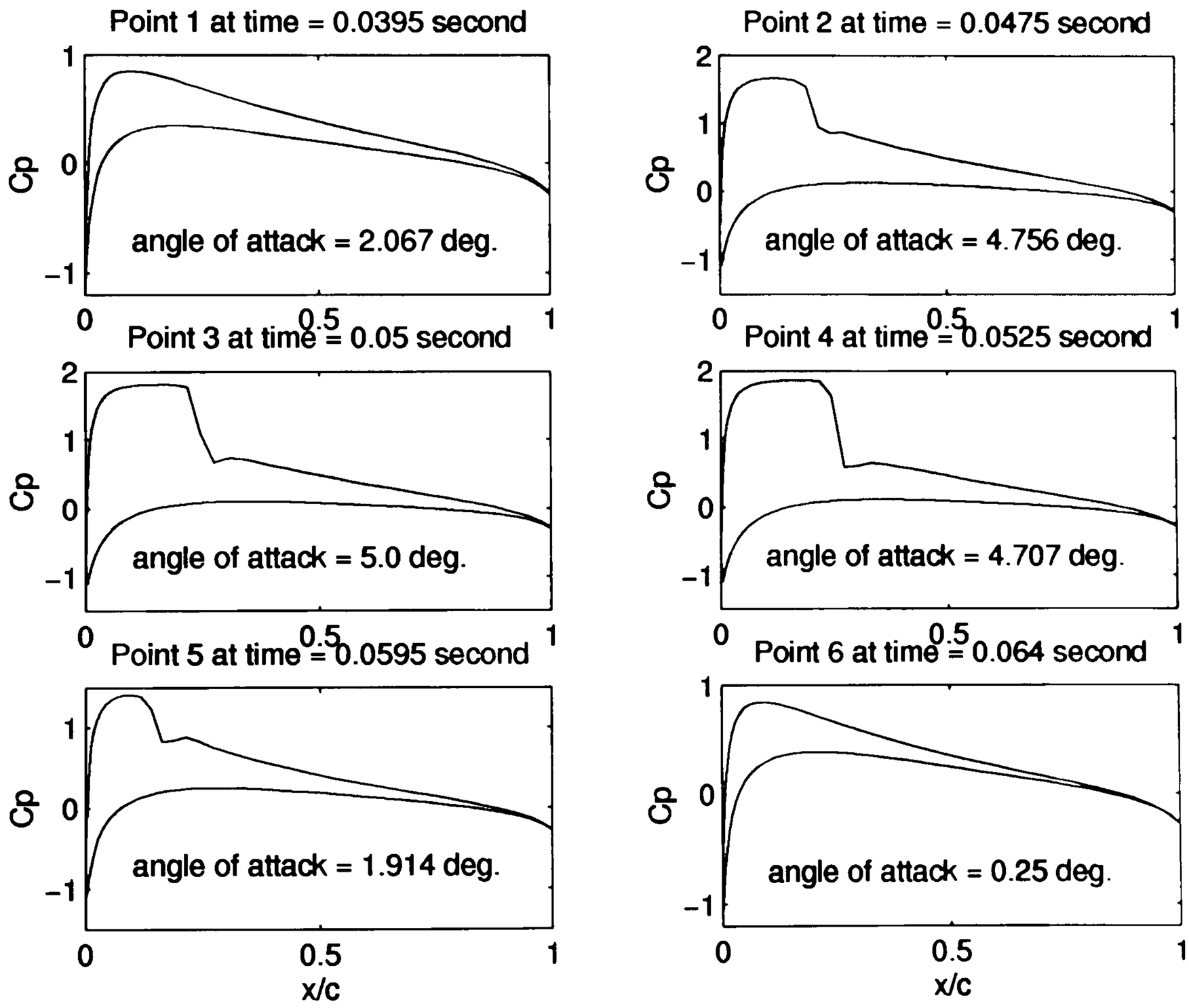
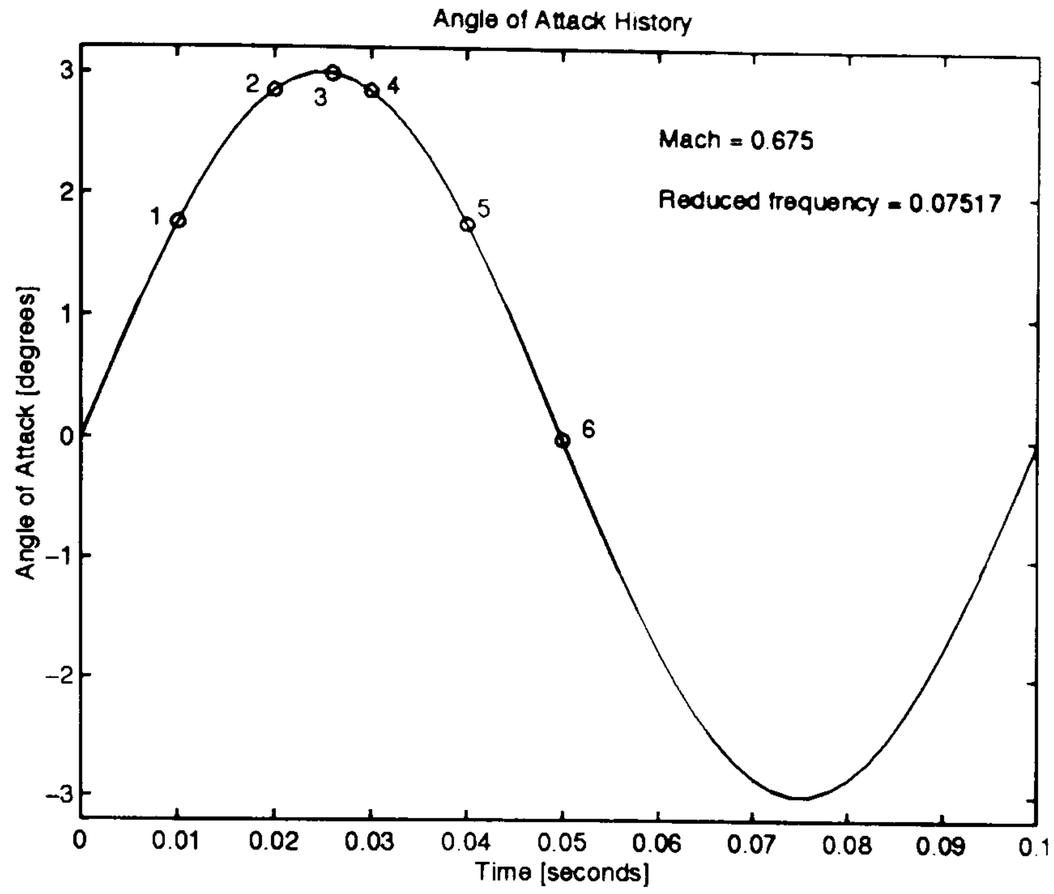


Figure B.15: Pressure distribution variation in relation to airfoil motion: Case in Figure 4.30 ($M = 0.65$).



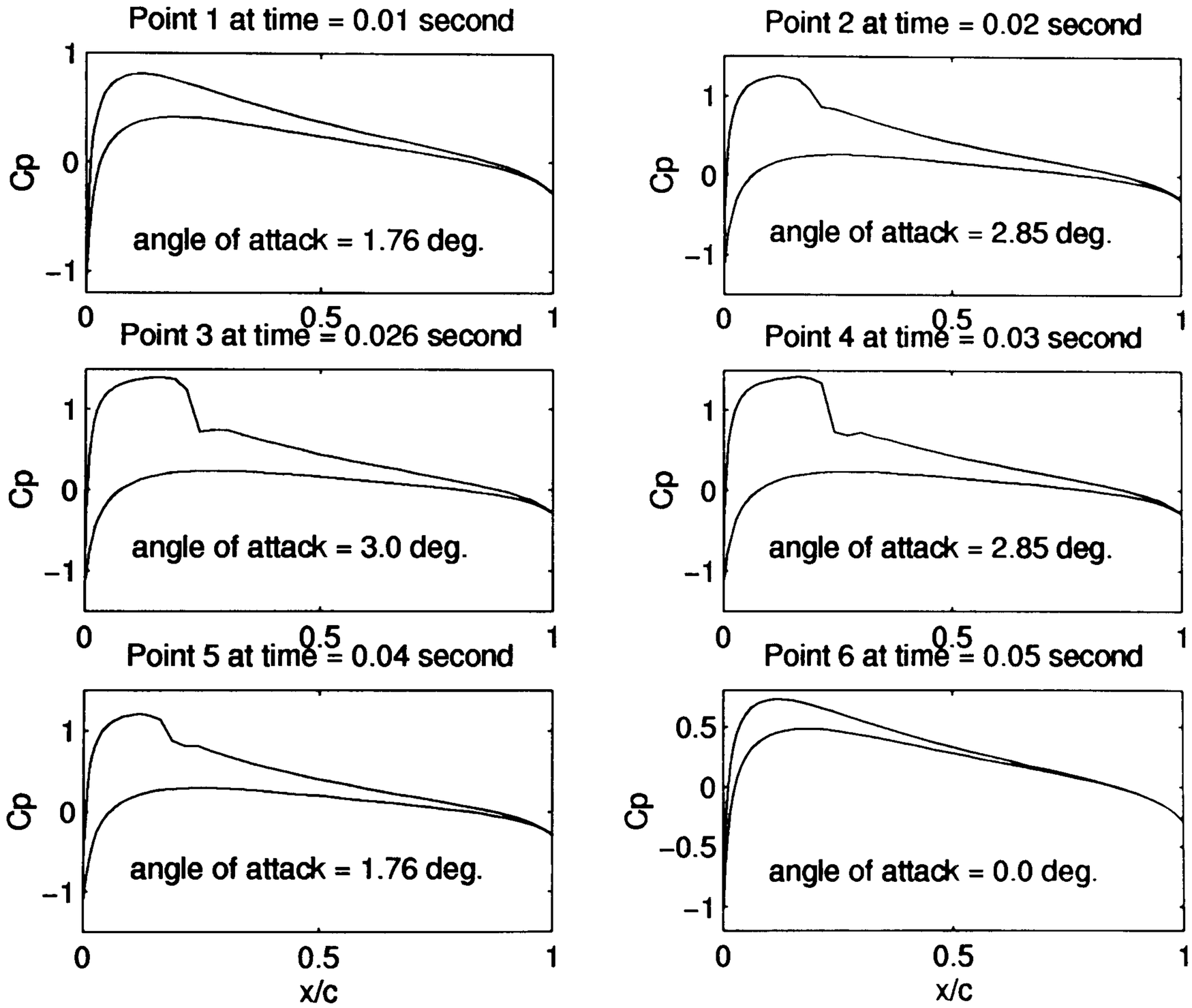
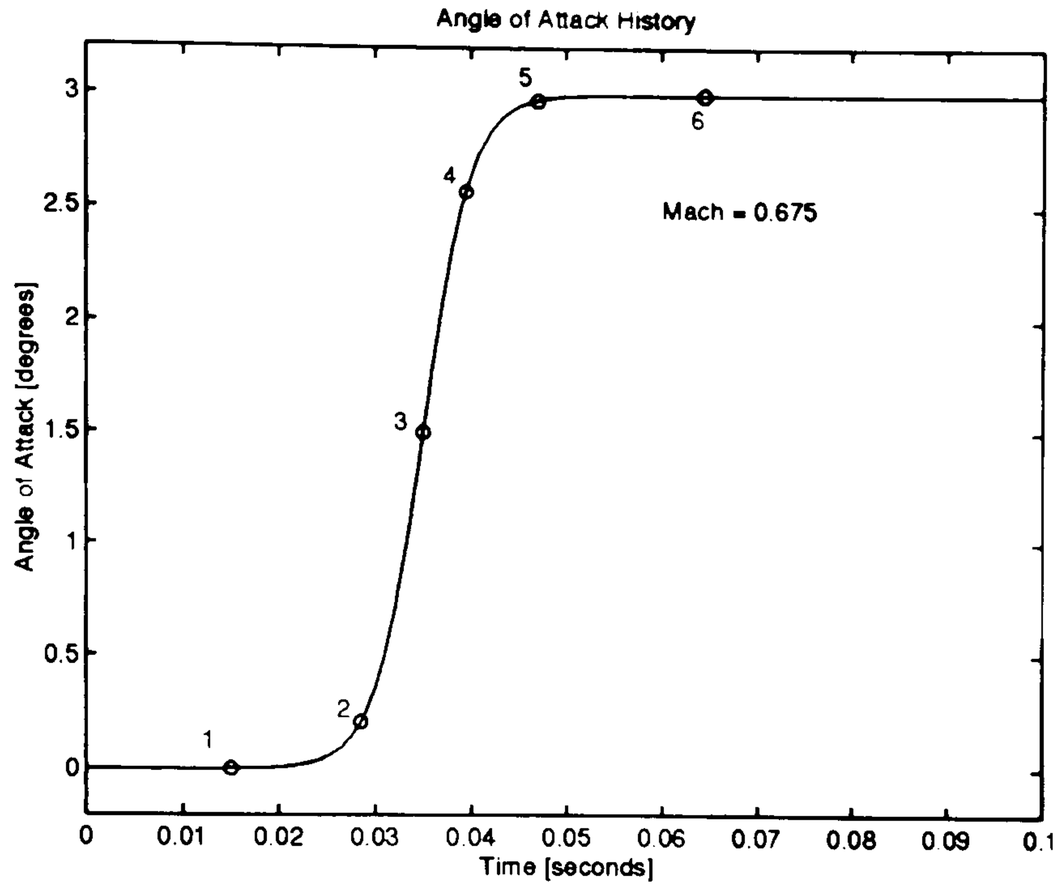


Figure B.16: Pressure distribution variation in relation to airfoil motion: Case in Figure 4.34 ($M = 0.675$).



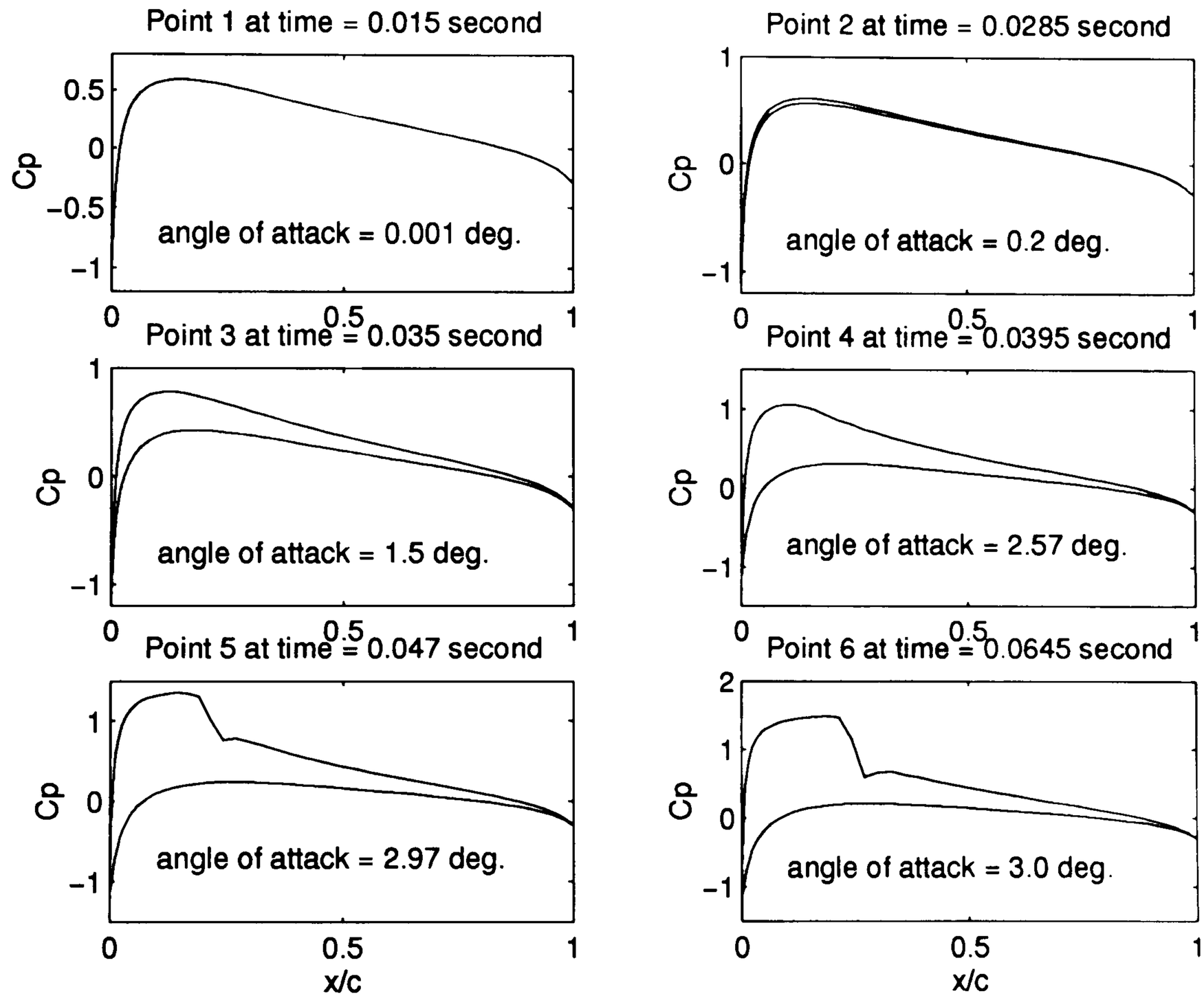
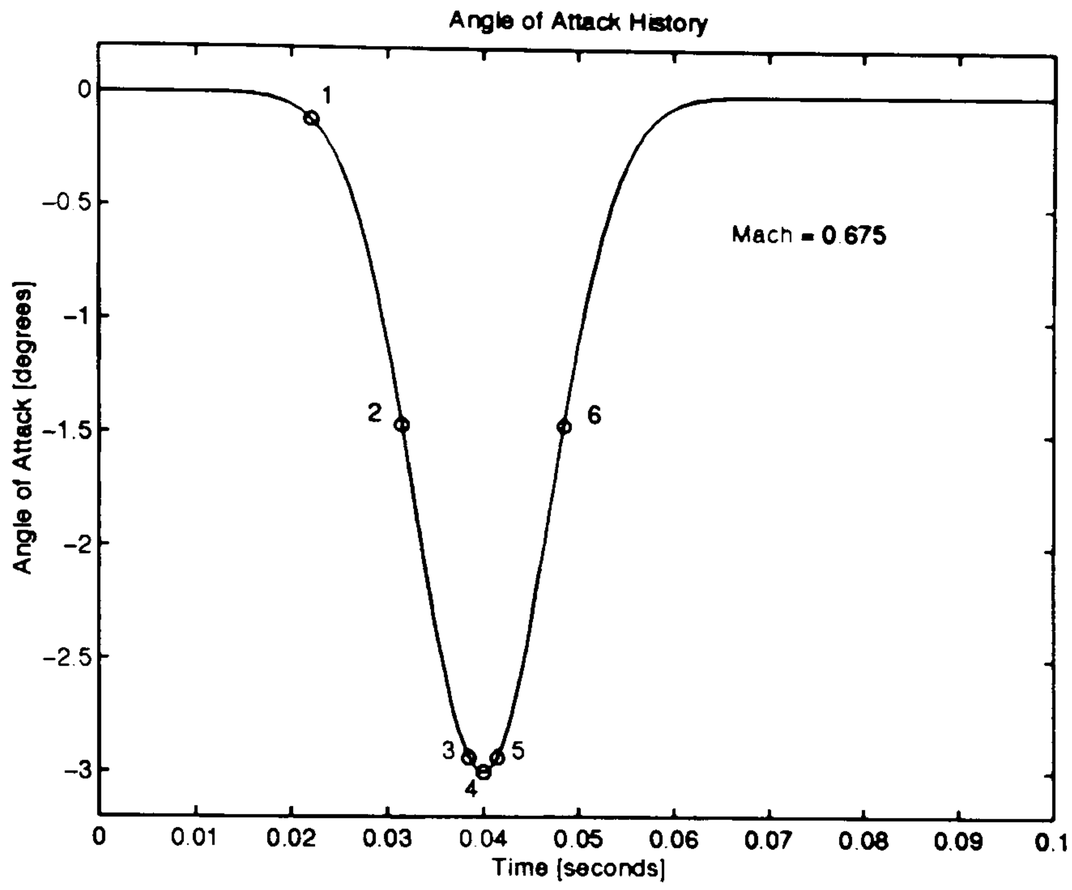


Figure B.17: Pressure distribution variation in relation to airfoil motion: Case in Figure 4.35 ($M = 0.675$).



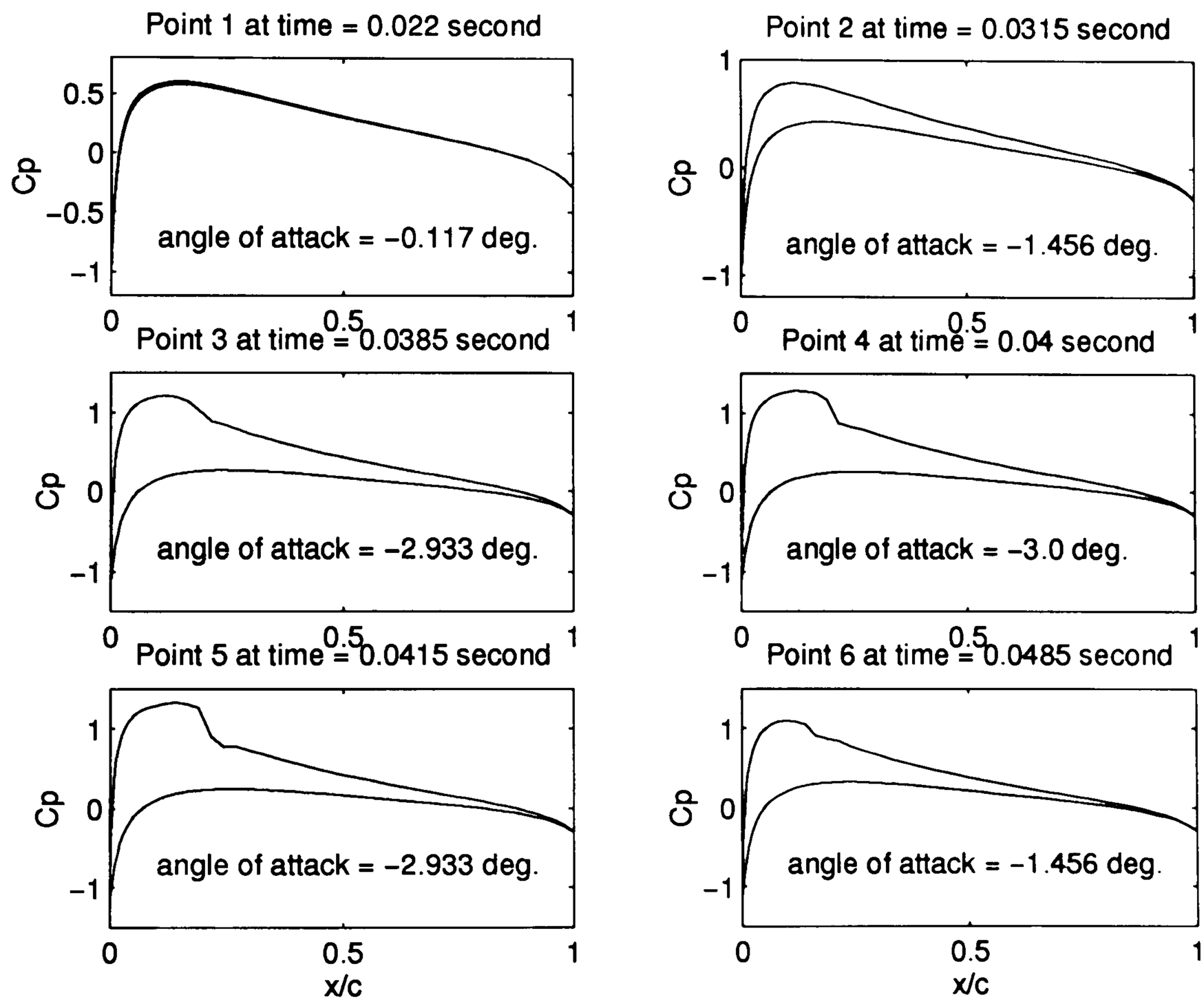
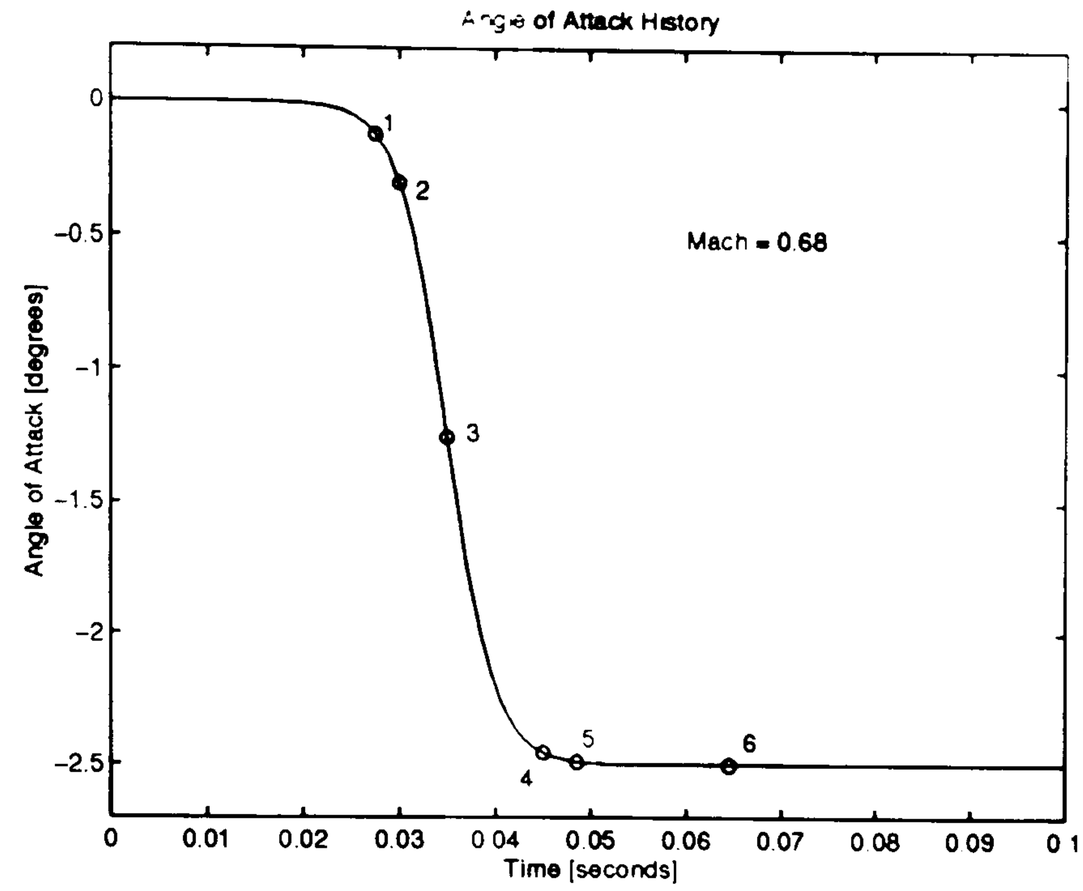


Figure B.18: Pressure distribution variation in relation to airfoil motion: Case in Figure 4.36 ($M = 0.675$).



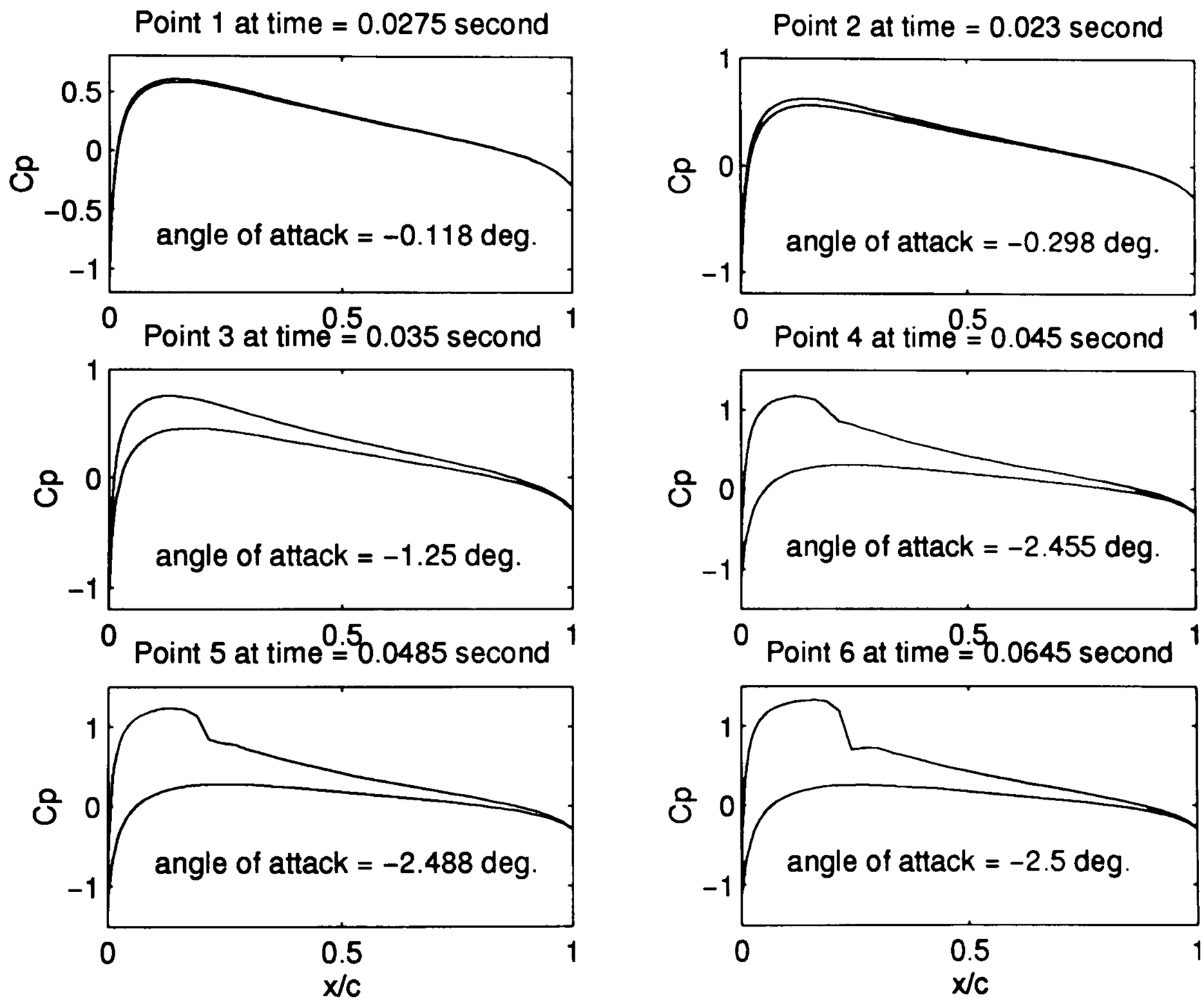
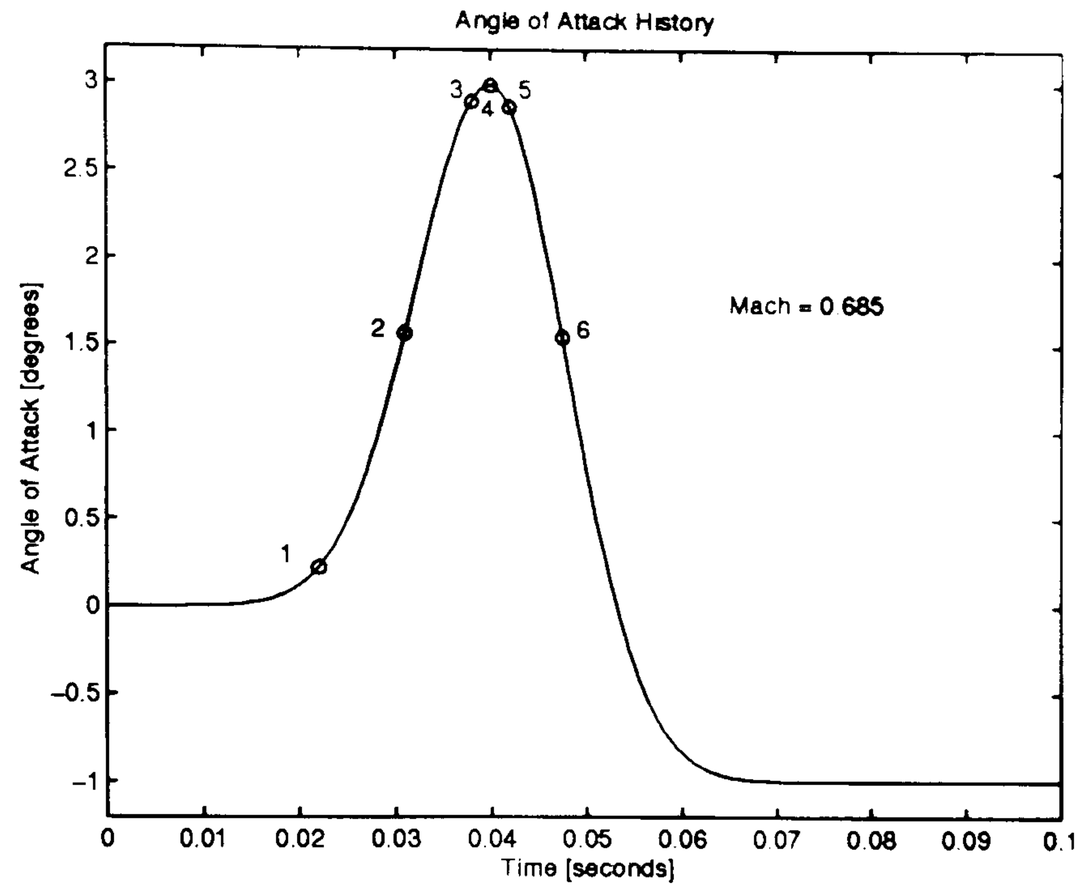


Figure B.19: Pressure distribution variation in relation to airfoil motion: Case in Figure 4.48 ($M = 0.68$).



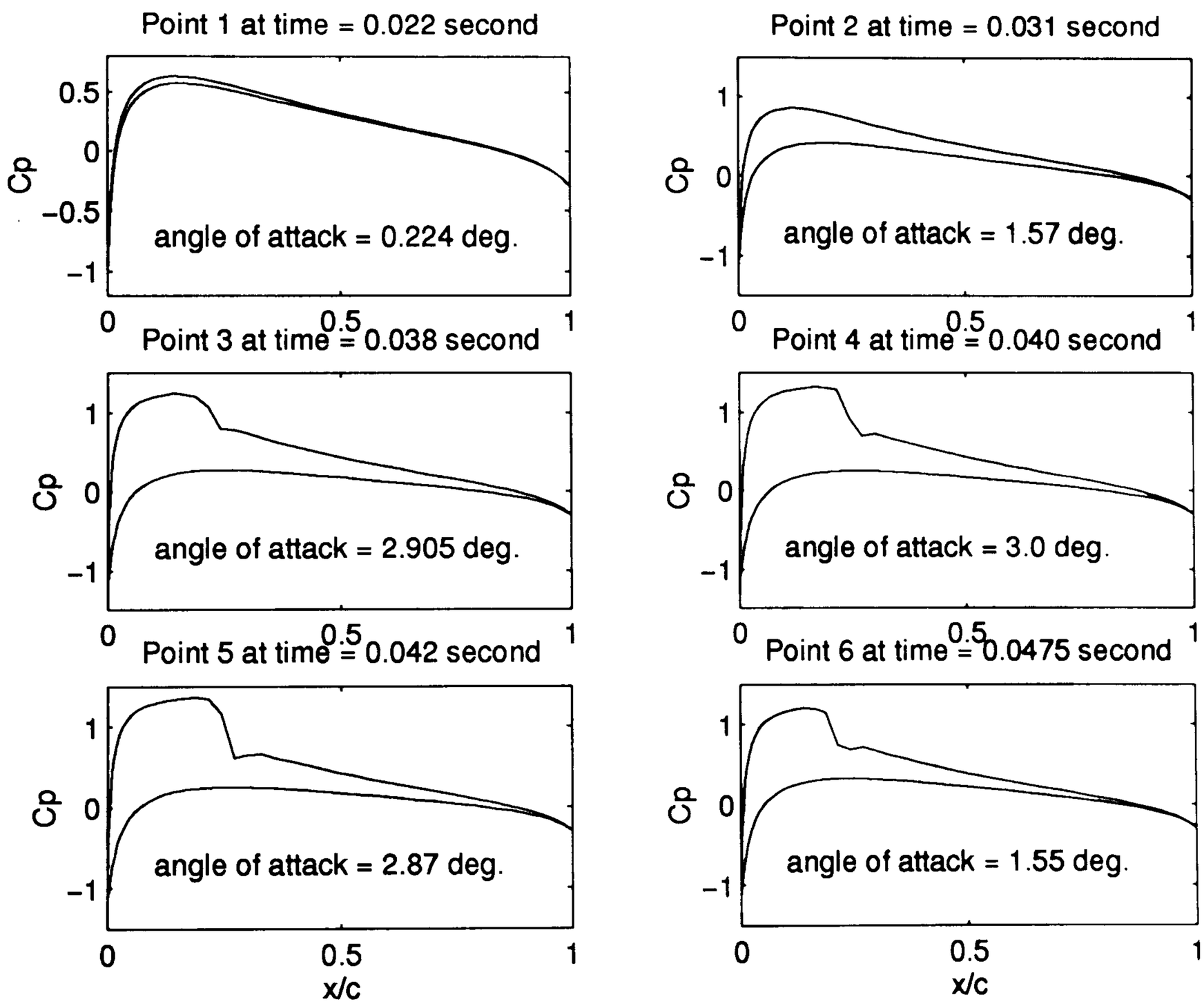
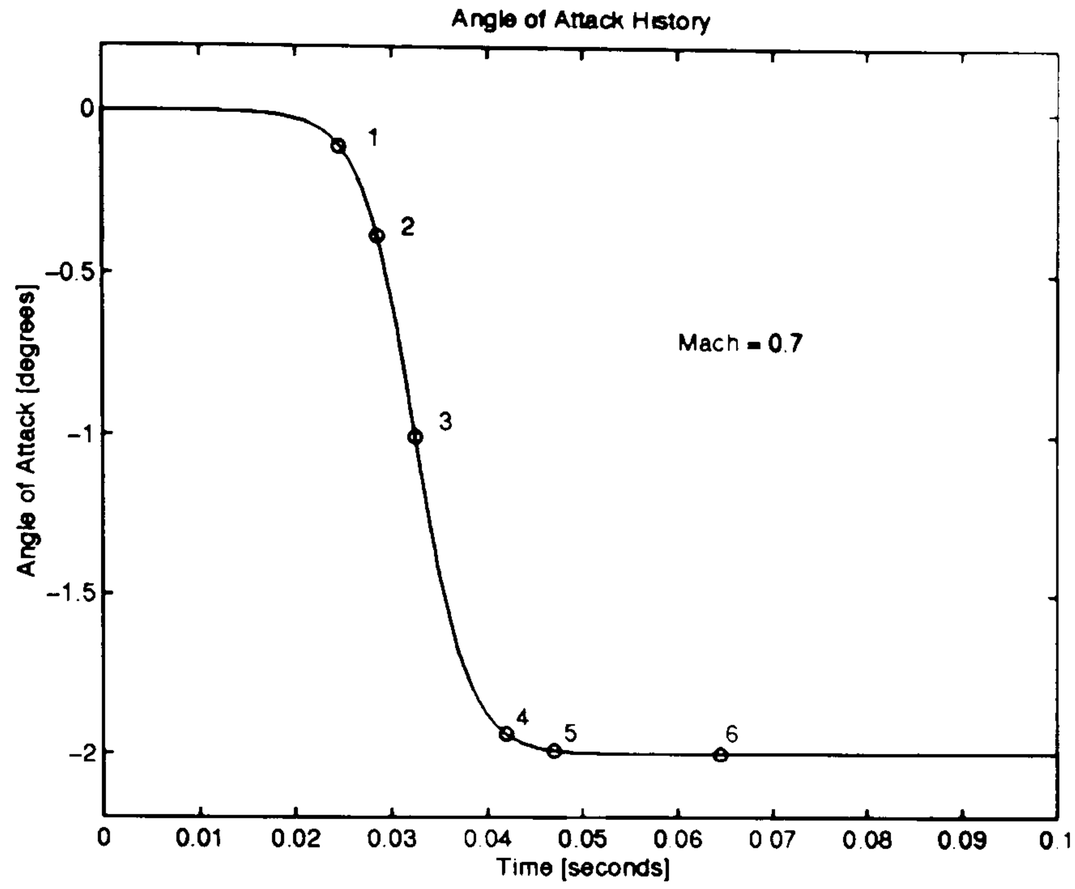


Figure B.20: Pressure distribution variation in relation to airfoil motion: Case in Figure 4.49 ($M = 0.685$).



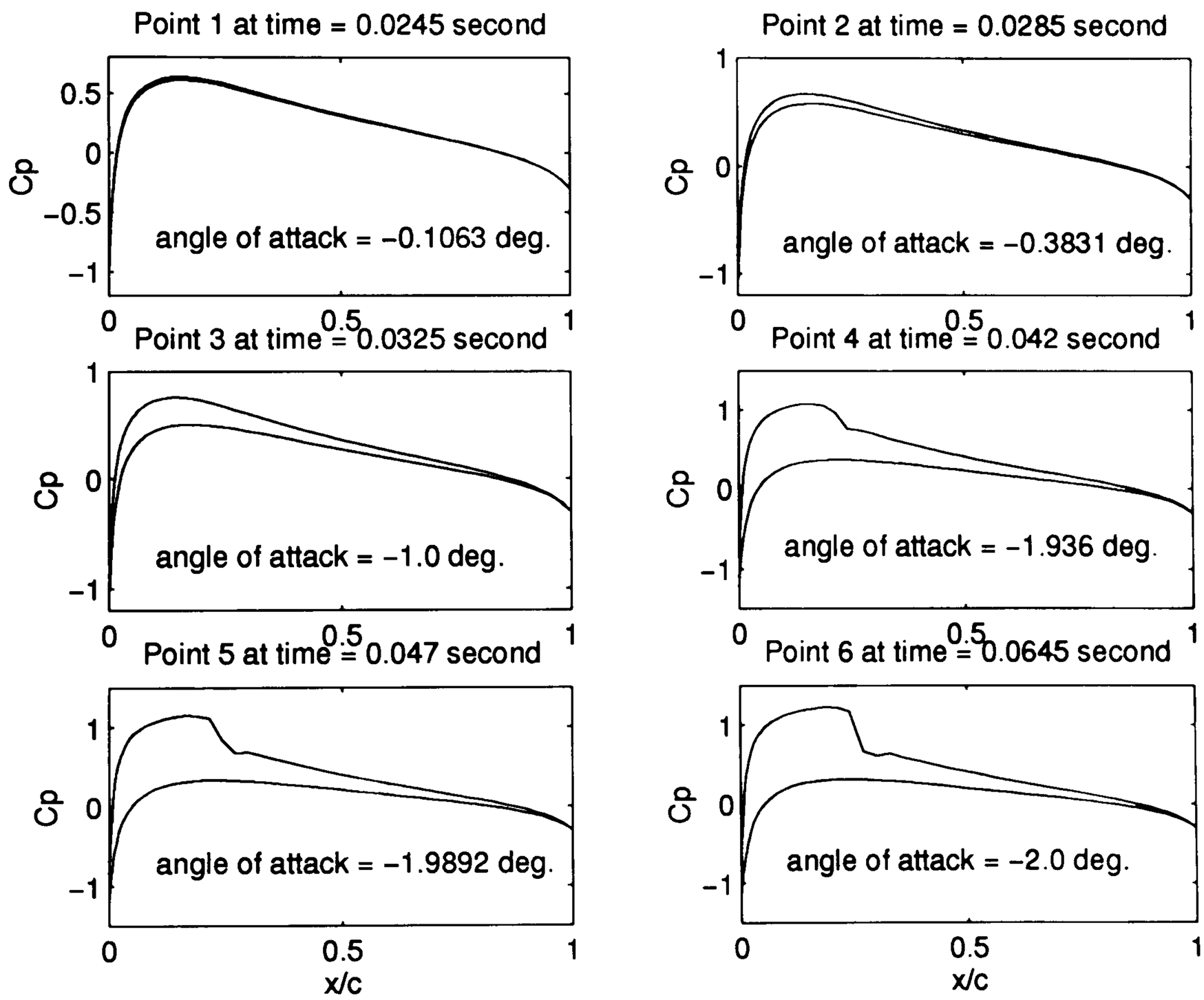
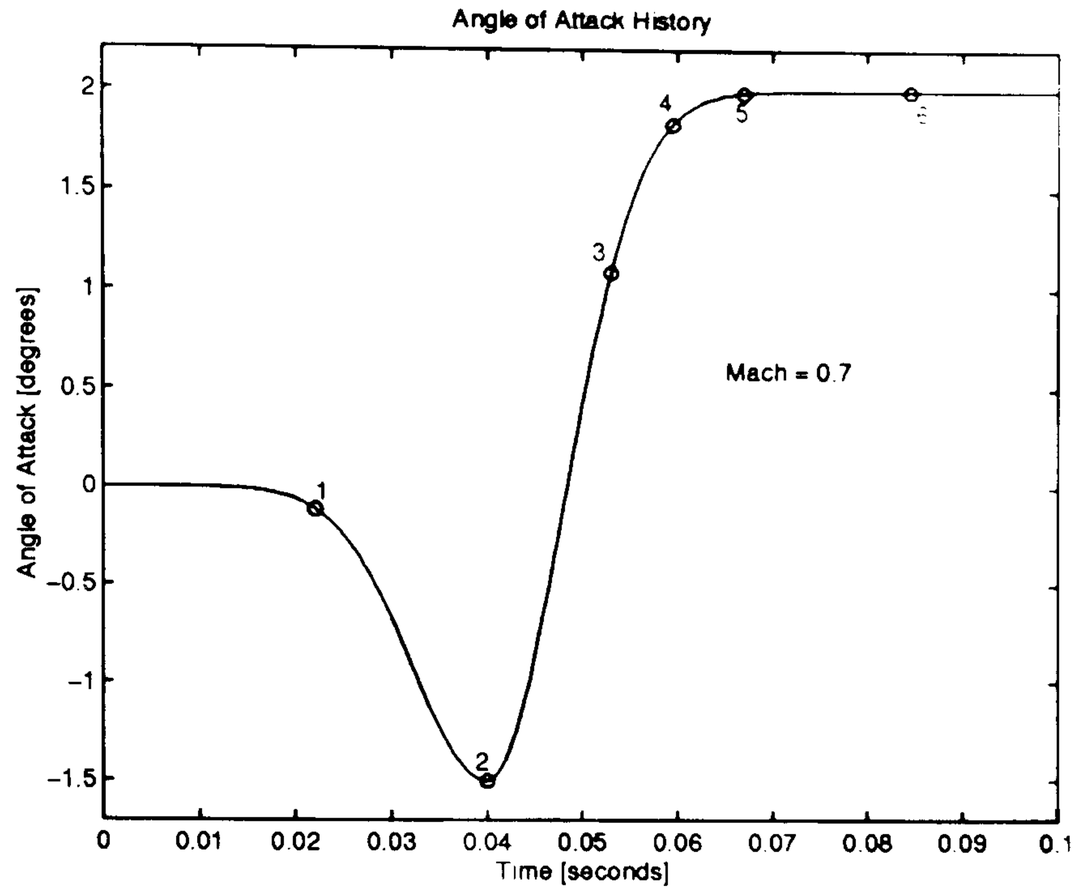


Figure B.21: Pressure distribution variation in relation to airfoil motion: Case in Figure 4.50 ($M = 0.7$).



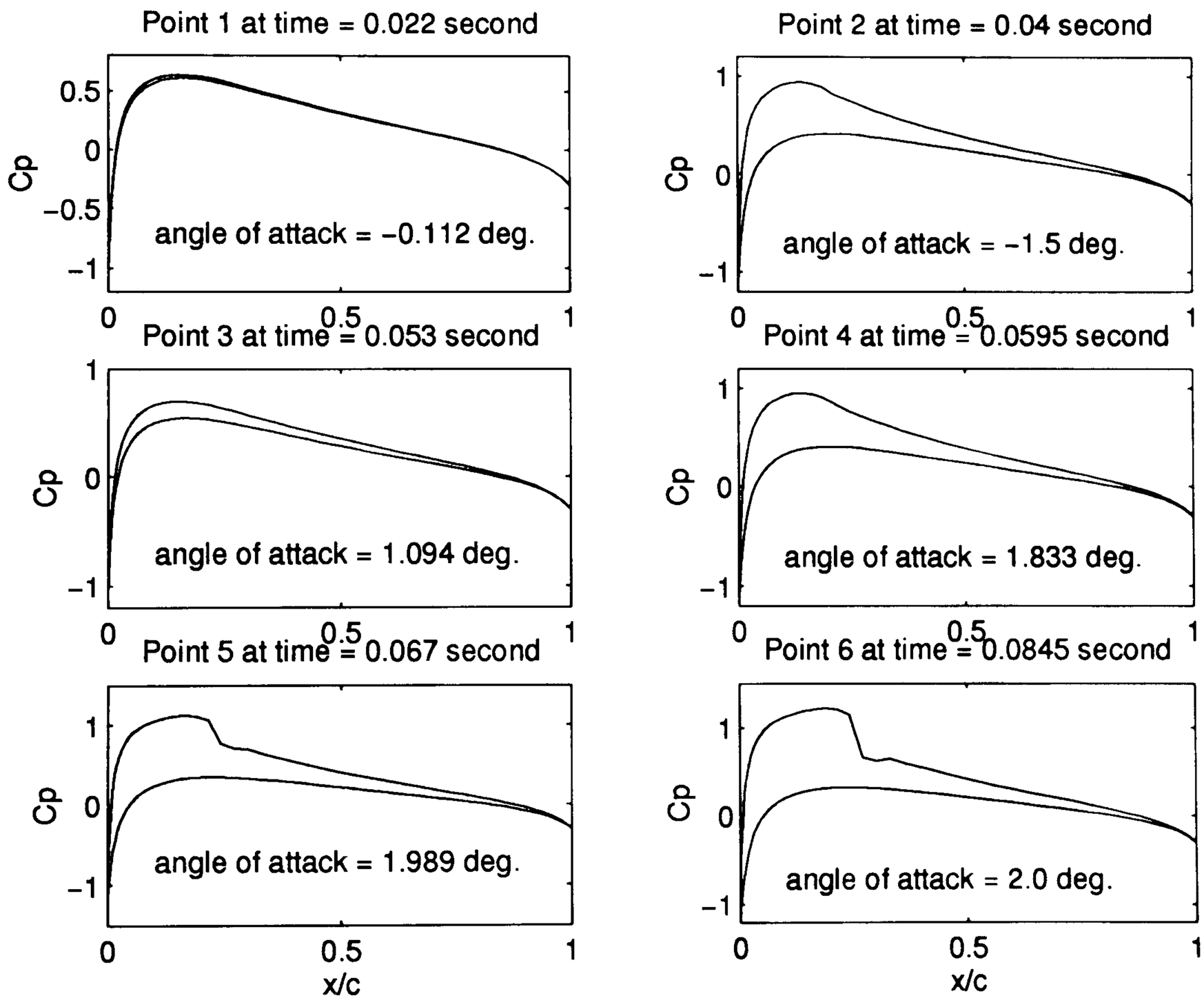
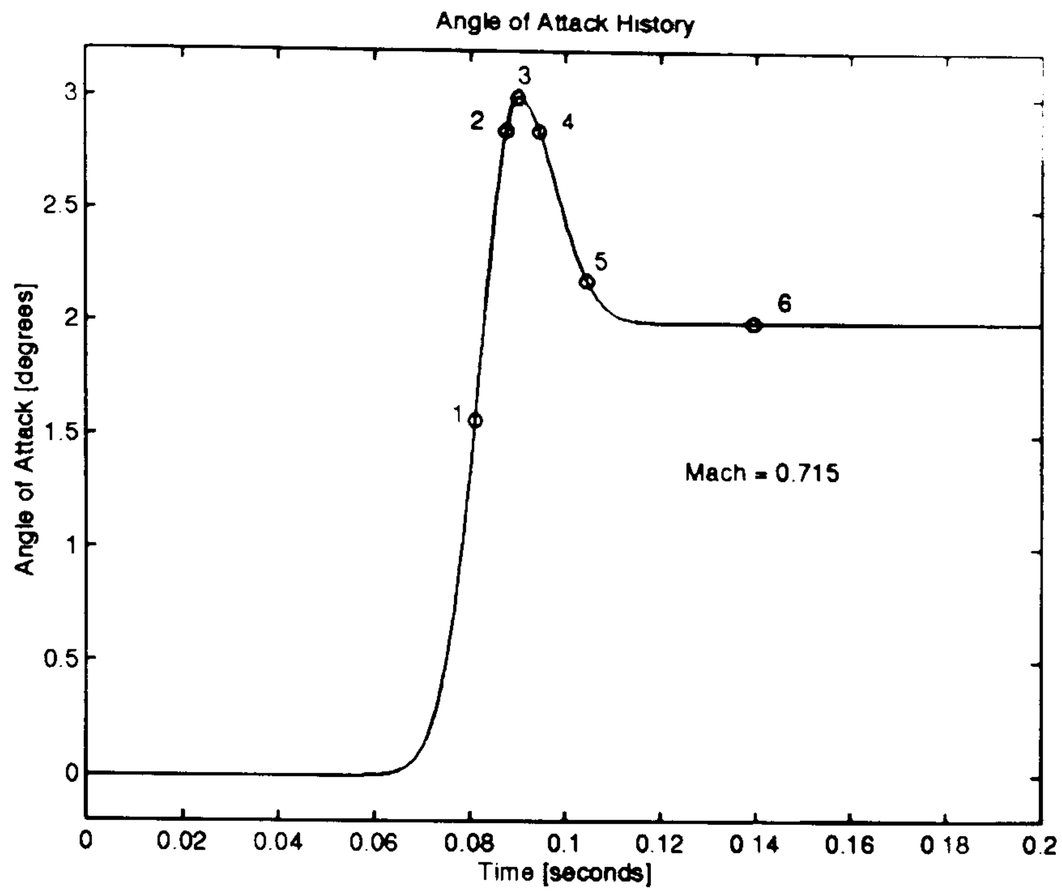


Figure B.22: Pressure distribution variation in relation to airfoil motion: Case in Figure 4.51 ($M = 0.7$).



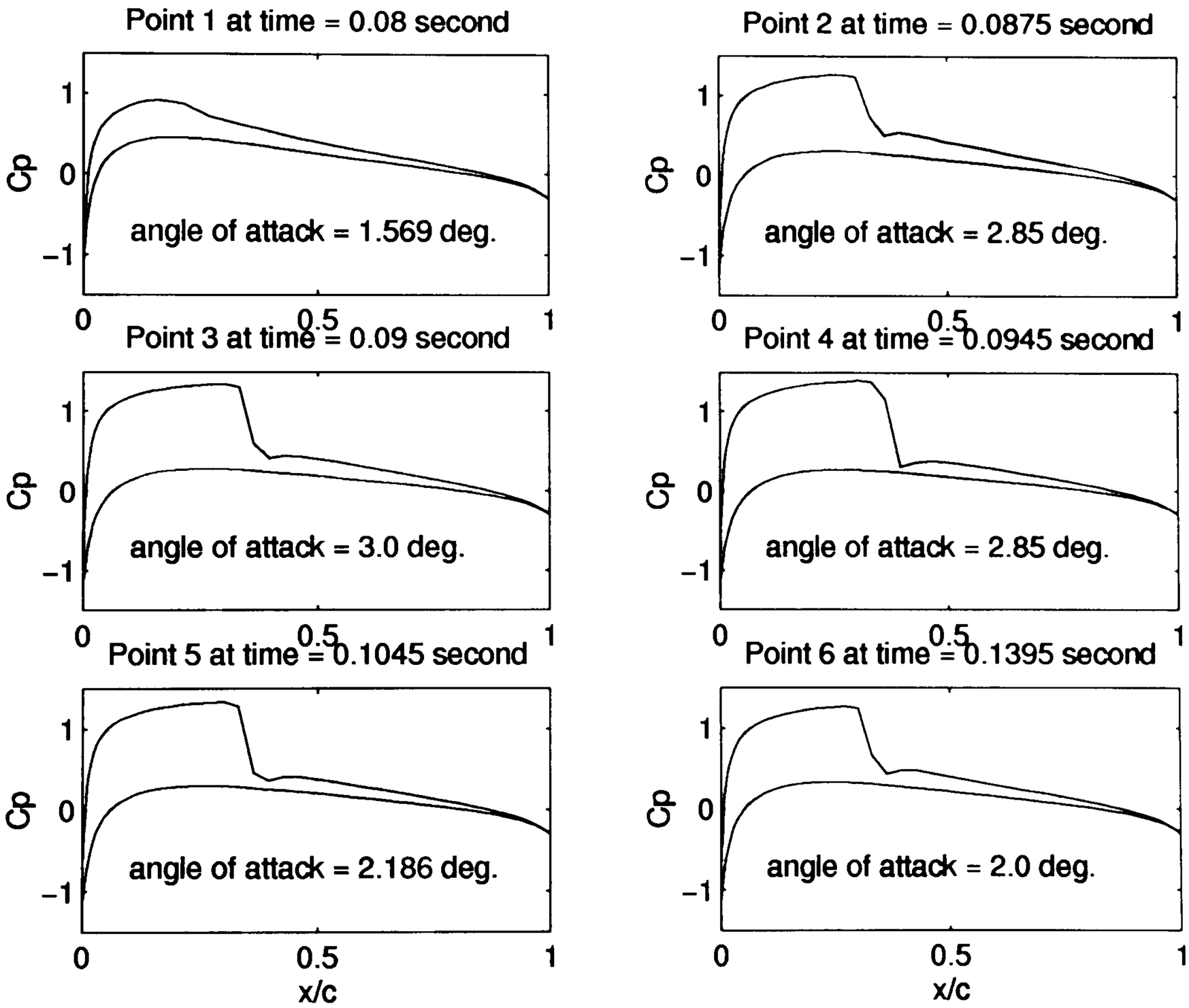
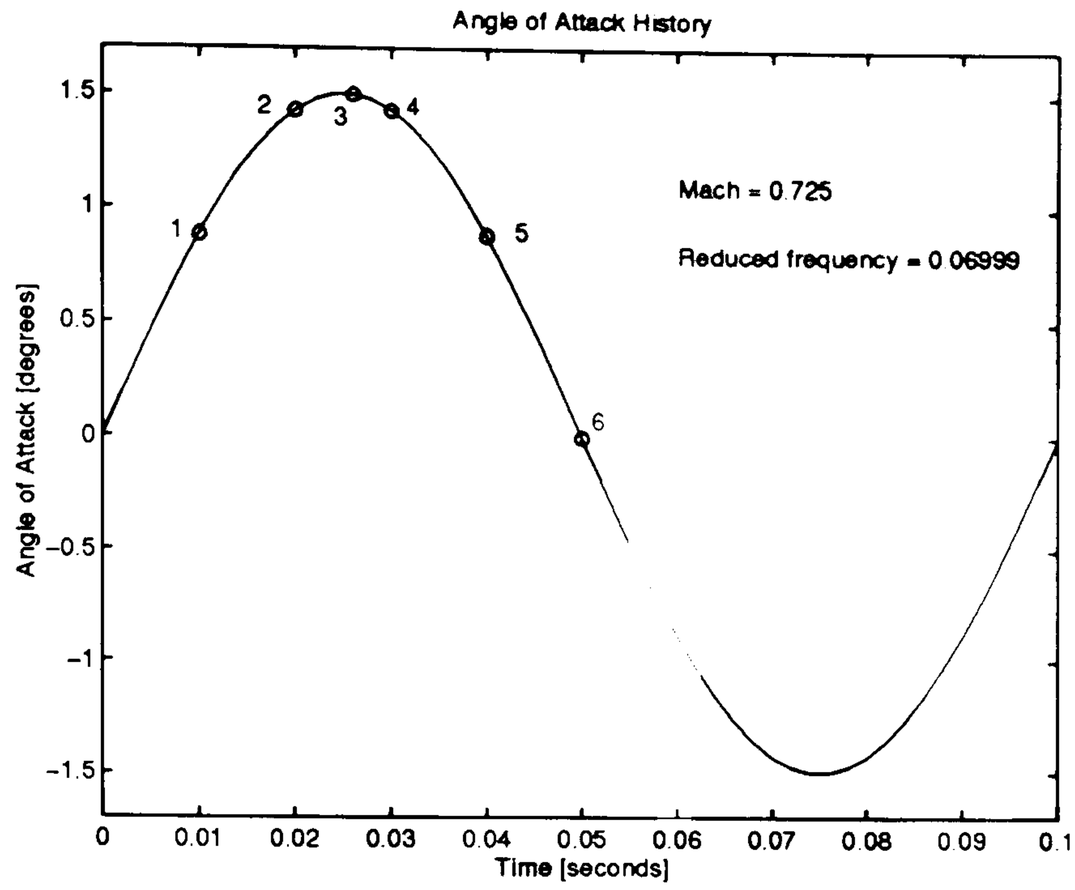


Figure B.23: Pressure distribution variation in relation to airfoil motion: Case in Figure 4.52 ($M = 0.715$).



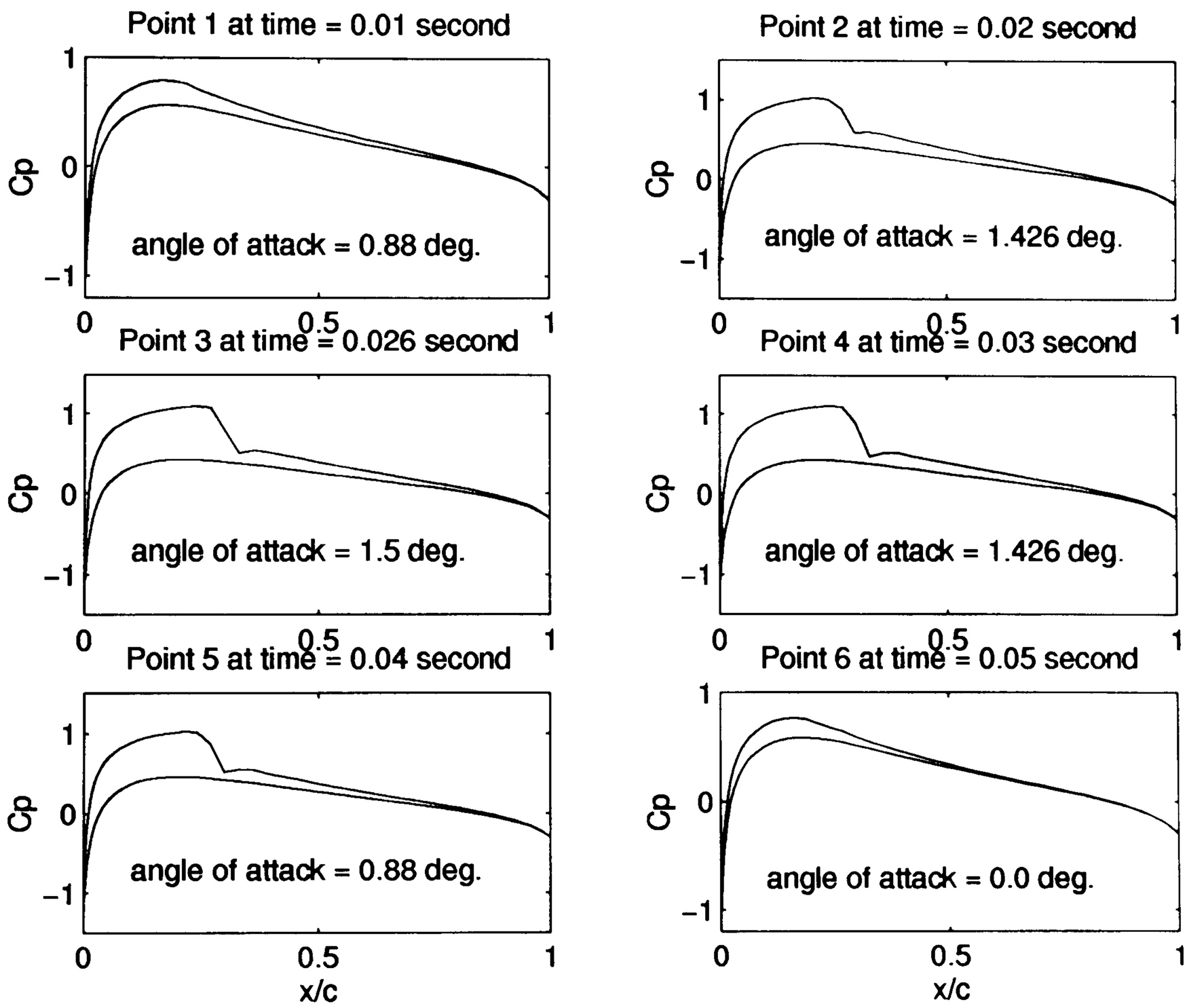
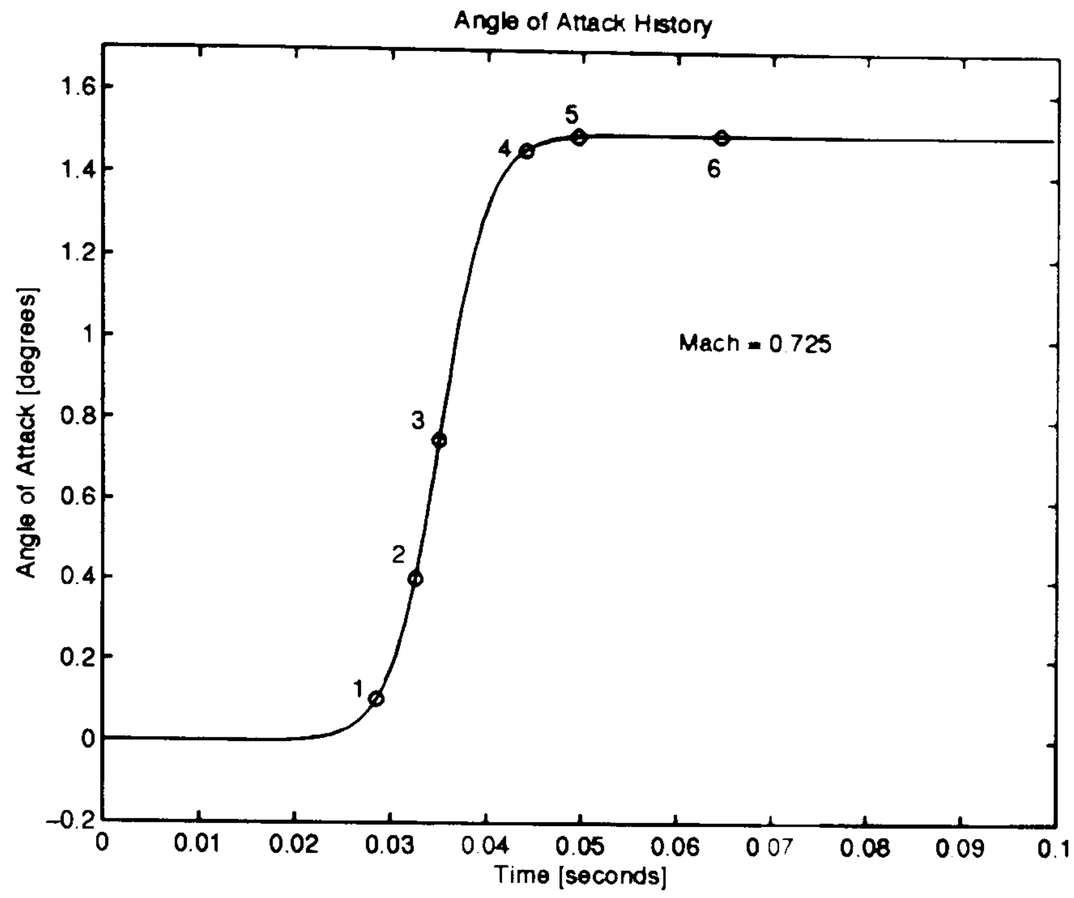


Figure B.24: Pressure distribution variation in relation to airfoil motion: Case in Figure 4.37 ($M = 0.725$).



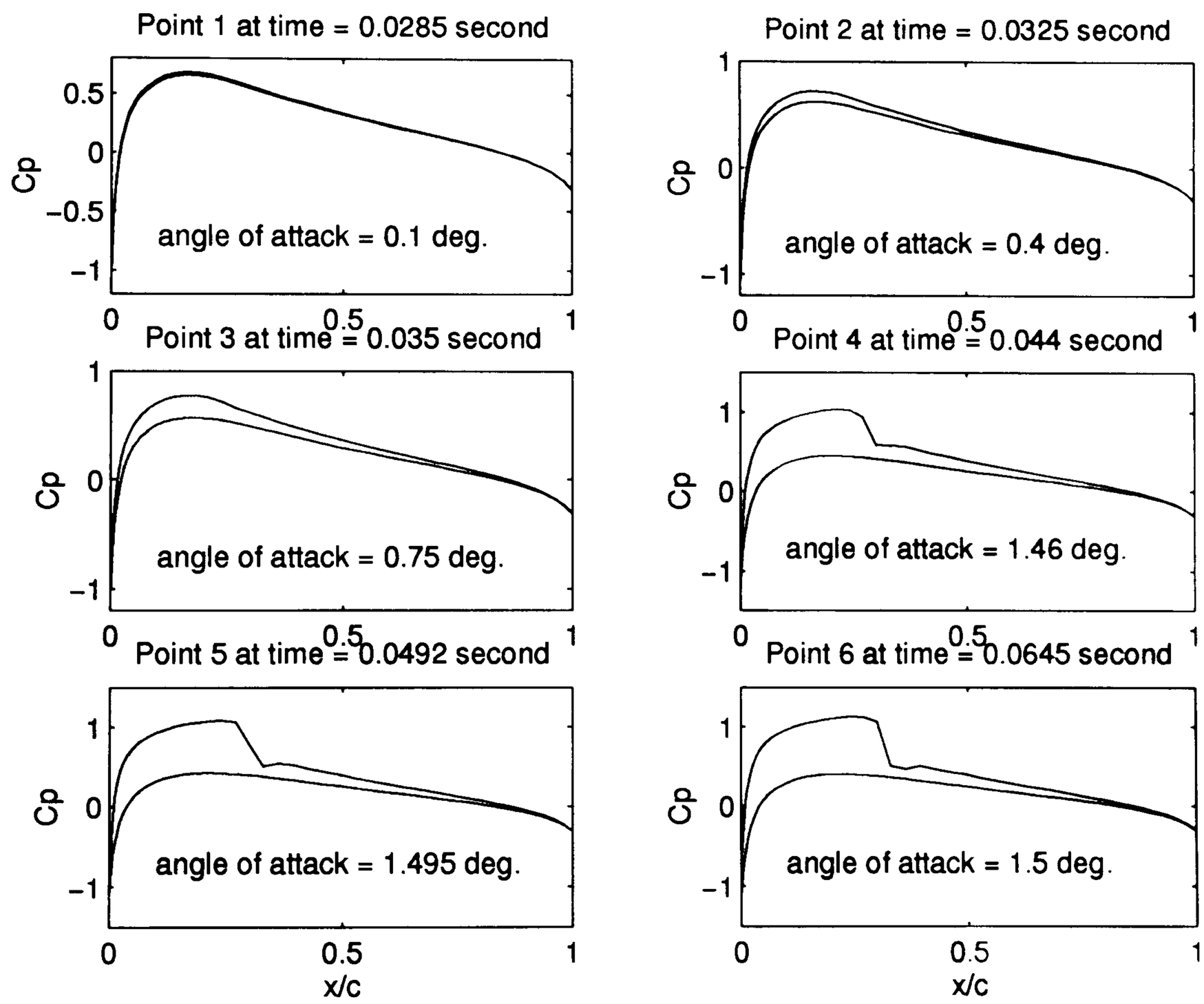
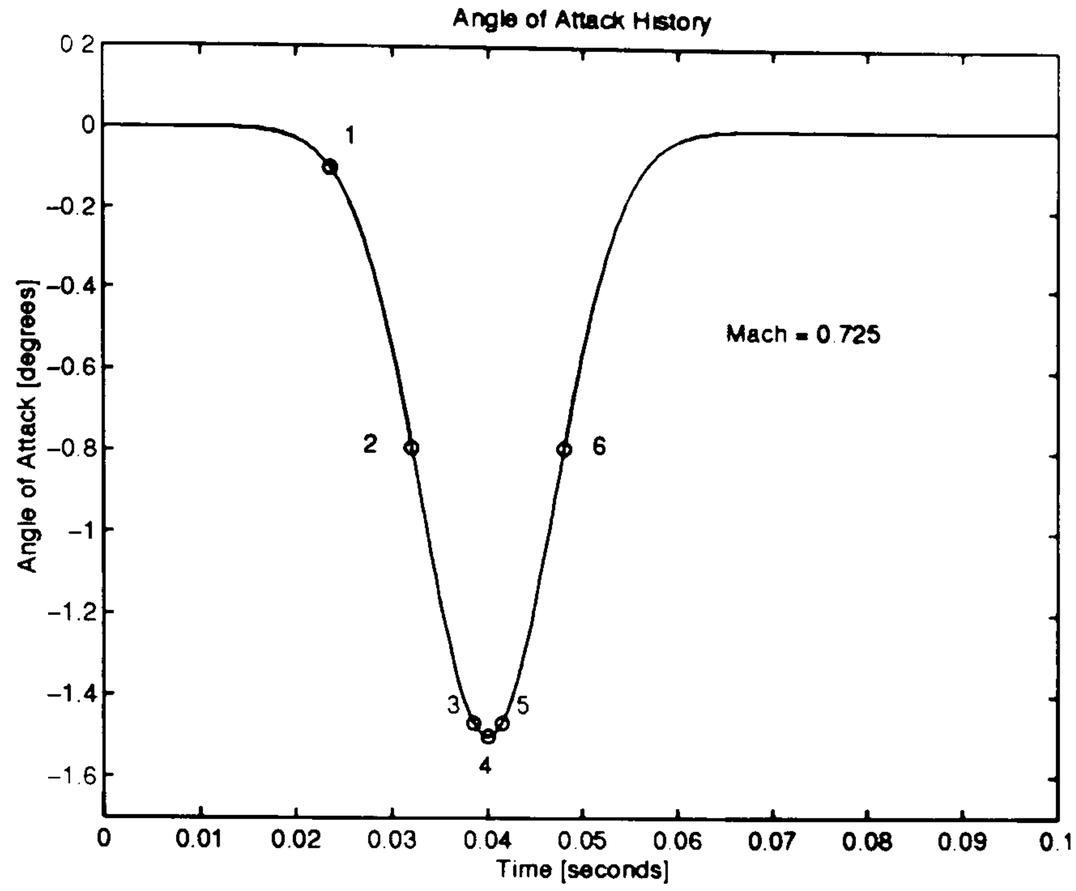


Figure B.25: Pressure distribution variation in relation to airfoil motion: Case in Figure 4.38 ($M = 0.725$).



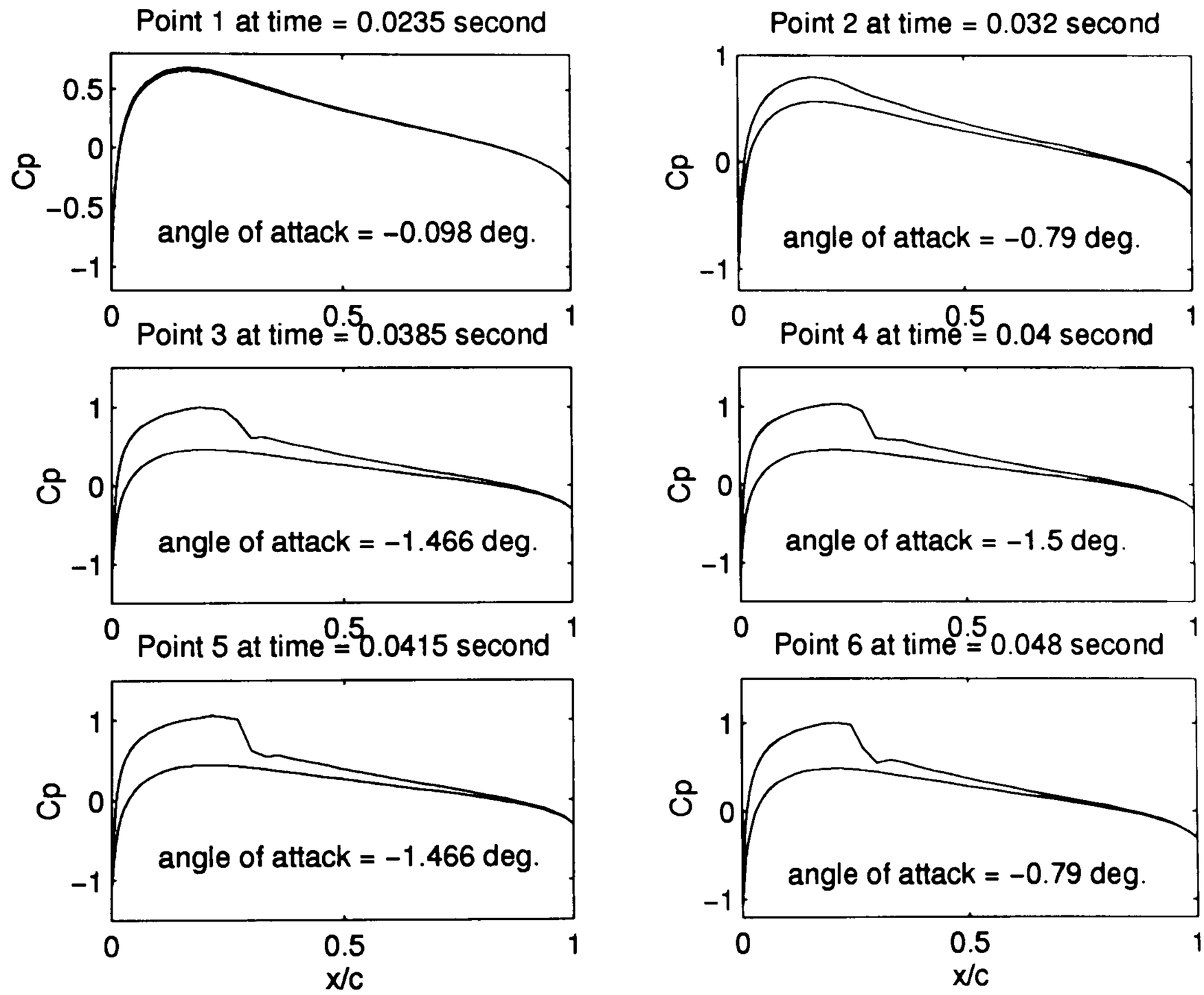
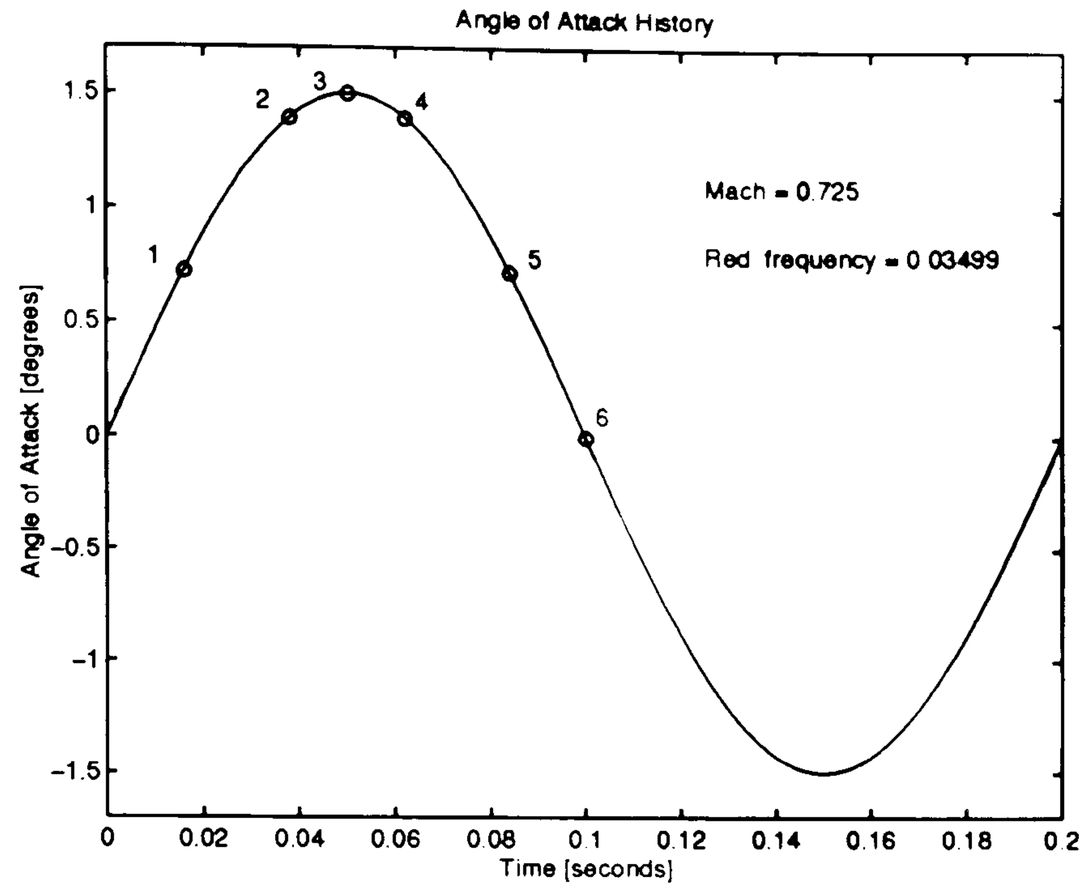


Figure B.26: Pressure distribution variation in relation to airfoil motion: Case in Figure 4.39 ($M = 0.725$).



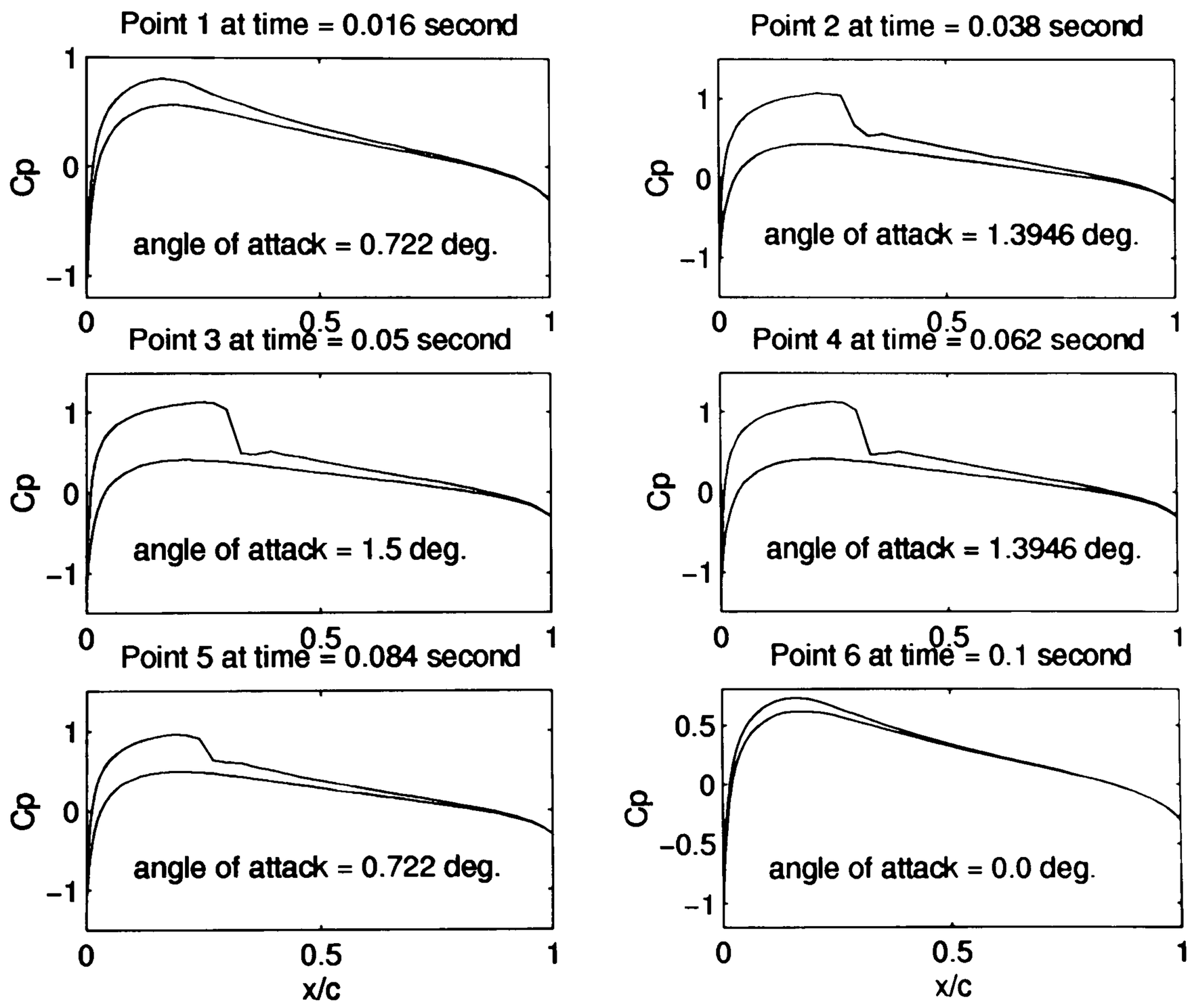
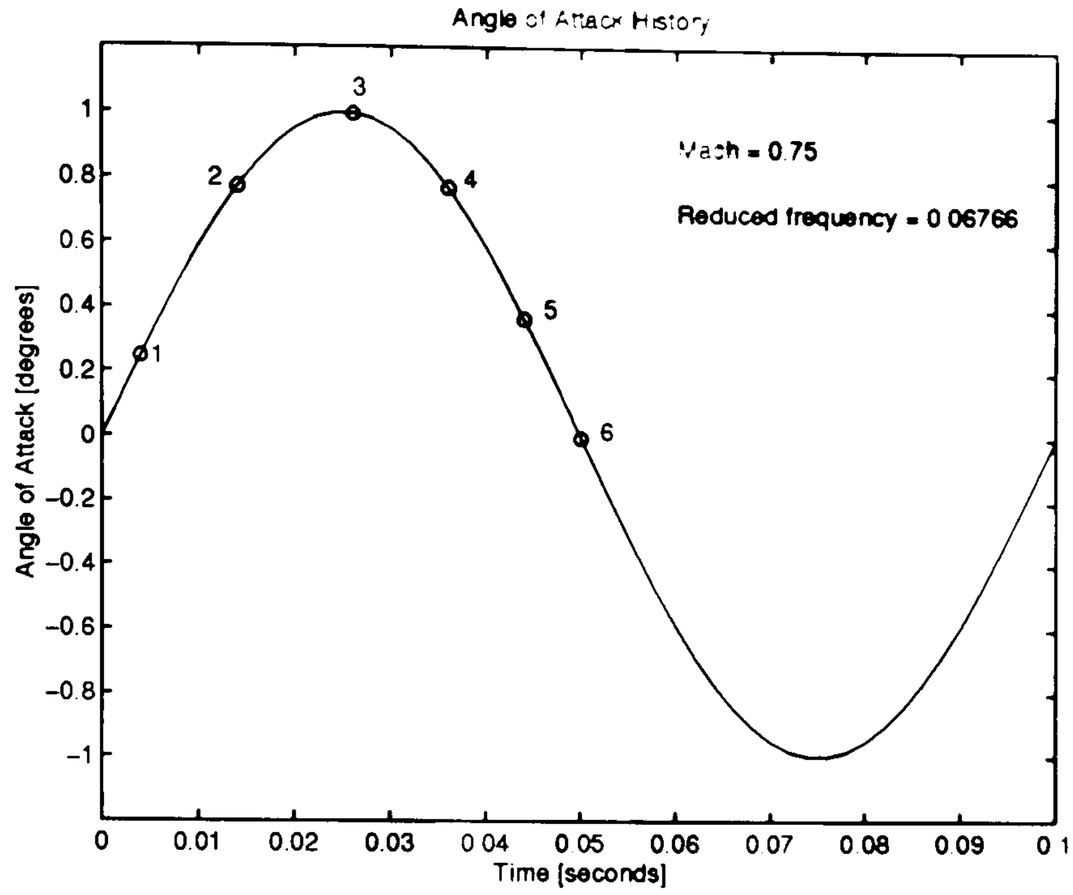


Figure B.27: Pressure distribution variation in relation to airfoil motion: Case in Figure 4.44 ($M = 0.725$).



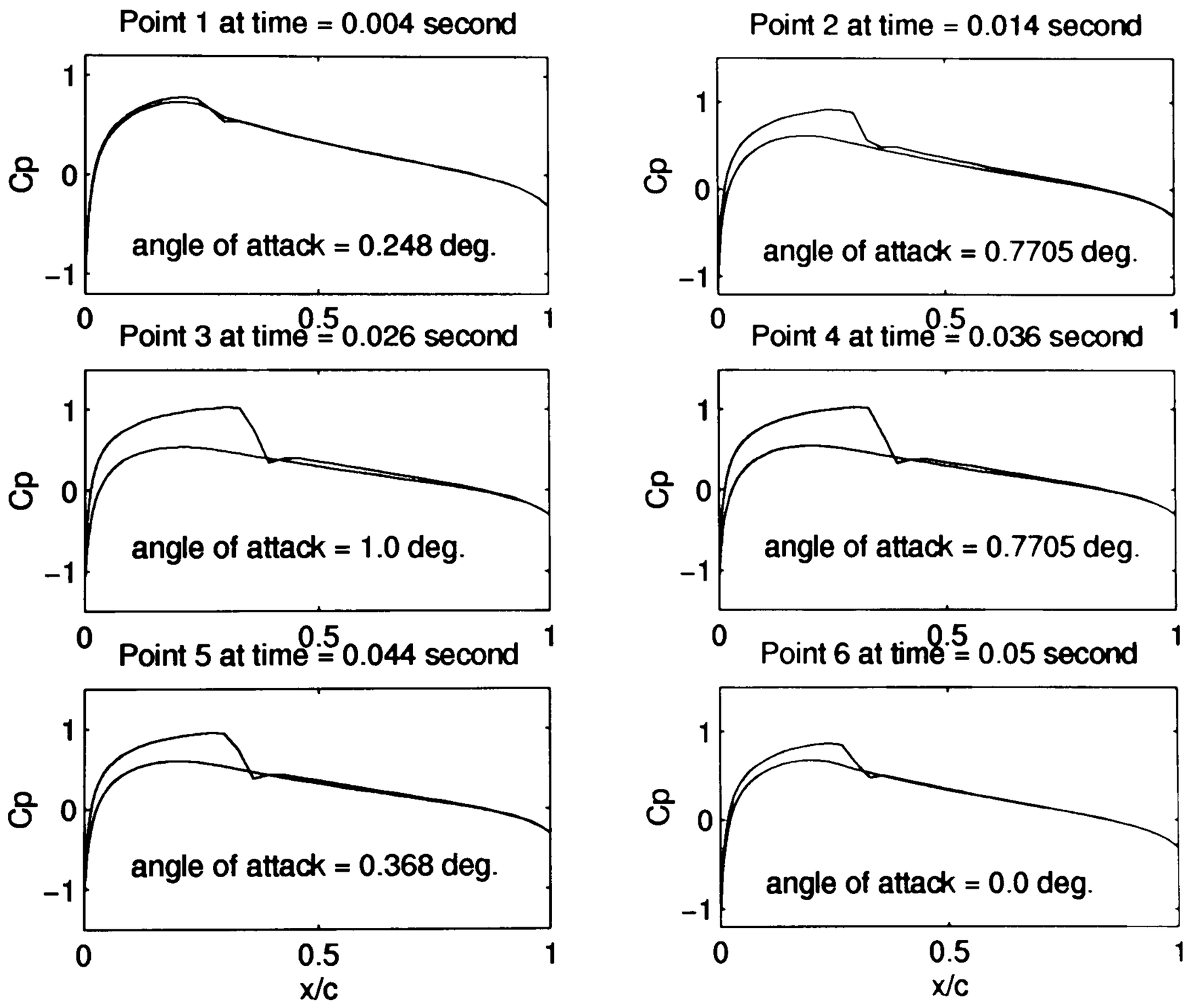


Figure B.28: Pressure distribution variation in relation to airfoil motion: Case in Figure 4.53 ($M = 0.75$).

Bibliography

[1] J. D. Anderson, Jr. *Fundamentals of Aerodynamics.* McGraw-Hill, Inc., New York, USA, second edition, 1991.

[2] P. J. Angeline, G. M. Saunders, and J. B. Pollack. An evolutionary algorithm that constructs recurrent neural networks. *IEEE Transactions on Neural Networks*, 5(1):54–65, January 1994.

[3] A. D. Back and A. C. Tsoi. FIR and IIR synapses, a new neural network architecture for time series modeling. *Neural Computation*, 3:375–385, 1991.

[4] A. D. Back, E. A. Wan, S. Lawrence, and A. C. Tsoi. A unifying view of some training algorithms for multilayer perceptron with FIR filter synapses. In *Fourth IEEE Workshop for Signal Processing*, pages 146–154, Ermoni, Greece, 6-8 September 1994.

[5] T. Bäck. *Evolutionary Algorithms in Theory and Practice.* Oxford University Press, New York, USA, 1996.

[6] D. Beasley, D. R. Bull, and R. R. Martin. An overview of genetic algorithms: Part 1, Fundamentals. *University Computing*, 15(2):58–69, 1993a.

[7] D. Beasley, D. R. Bull, and R. R. Martin. An overview of genetic algorithms: Part 2: Research topics. *University Computing*, 15(4):170–181, 1993b.

[8] T. S. Beddoes. A synthesis of unsteady aerodynamic effects including stall hysteresis. *Vertica*, 1:113–123, 1976.

[9] T. S. Beddoes. Practical computation of unsteady lift. In *Eighth European Rotorcraft Forum*, Aix-en-Provence, France, August 31 - September 3 1982a. Paper n. 2.3.

[10] T. S. Beddoes. Representation of airfoil behaviour. In *Specialists Meeting on Prediction of Aerodynamic Loads on Rotorcraft*, 1982b. AGARD CP-334.

[11] S. A. Billings. Identification of nonlinear systems - a survey. *IEE Proceedings D*, 127(6):272–285, 1980.

[12] S. A. Billings and S. Y. Fakhouri. Theory of separable processes with applications to the identification of nonlinear systems. *Proceedings of the IEE*, 125(9):1051–1058, October 1978.

[13] S. A. Billings and S. Y. Fakhouri. Nonlinear system identification using the Hammerstein model. *International Journal of Systems Science*, 10(5):567–578, 1979.

[14] S. A. Billings and S. Y. Fakhouri. Identification of systems containing linear dynamic and static nonlinear elements. *Automatica*, 18(1):15–26, 1982.

[15] R. L. Bisplinghoff and H. Ashley. *Principles of Aeroelasticity*. John Wiley and Sons, Inc., New York, USA, 1962.

[16] R. L. Bisplinghoff, H. Ashley, and R. L. Halfman. *Aeroelasticity*. Dover Publications, Inc., Mineola, New York, USA, 1996.

[17] T. Chen and H. Chen. Approximation of continuous functionals by neural networks with applications to dynamic systems. *IEEE Transactions on Neural Networks*, 4:910–918, 1993.

[18] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, 2:303–314, 1989.

[19] L. David, editor. *Handbook of Genetic Algorithms.* Van Nostrand Reinhold, New York, USA, 1991.

[20] S. P. Day and D. S. Camponese. Continuous-time temporal back-propagation. In *1991 International Joint Conference on Neural Networks*, pages 95–100, Seattle, WA, USA, July 1991.

[21] S. P. Day and M. R. Davenport. Continuous-time temporal back-propagation with adaptable time delays. *IEEE Transactions on Neural Networks*, 4(2):348–354, March 1993.

[22] E. H. Dowell. Nonlinear aeroelasticity. In A. K. Noor and S. L. Venneri, editors, *Structural Dynamics and Aeroelasticity*, volume 5 of *Flight-Vehicle Materials, Structures, and Dynamics - Assessment and Future Directions*, pages 213–239. The American Society of Mechanical Engineers, 1993. Chapter 4.

[23] E. H. Dowell, H. C. Curtiss, Jr., R. H. Scalan, and F. Sisto. *A Modern Course in Aeroelasticity.* Kluwer Academic Publishers, Dordrecht, The Netherlands, second revised and enlarged edition, 1989.

[24] E. H. Dowell and M. Ilgamov. *Studies in Nonlinear Aeroelasticity.* Springler-Verlag, New York, USA, 1988.

[25] L. Dubuc, F. Cantarini, M. Woodgate, B. Gribben, K. J. Badcock, and B. E. Richards. Solution of the Euler unsteady equations using deforming grids. University of Glasgow, Report GU-AERO-9704, February 1997.

[26] J. W. Edwards and J. L. Thomas. Computational methods for unsteady transonic flows. In D. Nixon, editor, *Unsteady Transonic Aerodynamics*, volume 120 of *Progress in Astronautics and Aeronautics*, pages 211–261. AIAA, 1989.

[27] L. E. Ericsson and J. P. Reding. Fluid dynamics of unsteady separated flow. Part II. Lifting surfaces. *Progress in Aerospace Sciences*, 24:249–356. 1987.

[28] B. Etkin and L. D. Reid. *Dynamics of Flight: Stability and Control.* John Wiley & Sons, Inc., USA, third edition, 1996.

[29] W. E. Faller and S. J. Schreck. Real-time prediction of unsteady aerodynamics: Application for aircraft control and maneuverability enhancement. *IEEE Transactions on Neural Networks*, 6(6):1461–1468, November 1995.

[30] W. E. Faller and S. J. Schreck. Neural networks: Applications and opportunities in aeronautics. *Progress in Aerospace Science*, 32:433–456, 1996.

[31] W. E. Faller and S. J. Schreck. Unsteady fluid mechanics applications of neural networks. *Journal of Aircraft*, 34(1):48–55, January-February 1997.

[32] D. B. Fogel, L. J. Fogel, and V. W. Porto. Evolving neural networks. *Biological Cybernetics*, 63:487–493, 1990.

[33] Y. C. Fung. *An Introduction to the Theory of Aeroelasticity.* Dover Publications, Inc., Mineola, New York, USA, 1993.

[34] D. E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley, 1989.

[35] S. A. Harp and T. Samad. Genetic synthesis of neural networks architecture. In L. David, editor, *Handbook of Genetic Algorithms*, New York, USA, 1991. Van Nostrand Reinhold.

[36] S. Haykin. *Neural Networks - A Comprehensive Foundation.* Macmillan College Publishing Company, Inc., New York, USA, 1994.

[37] R. Hecht-Nielsen. *Neurocomputing.* Addison-Wesley Publishing Company, Inc., Reading, USA, 1990.

[38] J. Hertz, A. Krogh, and R. G. Palmer. *Introduction to the Theory of Neural Computation.* Addison-Wesley, Reading, MA, 1991.

[39] J. H. Holland. Genetic algorithms. *Scientific American*, 267(1):44–50, July 1992.

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

[41] I. W. Hunter and M. J. Korenberg. The identification of nonlinear biological systems: Wiener and Hammerstein cascade models. *Biological Cybernetics*, 55:135–144, 1986.

[42] J. E. Jenkins. Simplification of nonlinear indicial response models: Assessment of the two-dimensional airfoil case. *Journal of Aircraft*, 28(2):131–138, February 1991.

[43] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, May 1983.

[44] M. J. Korenberg and I. W. Hunter. The identification of nonlinear biological systems: LNL cascade models. *Biological Cybernetics*, 55:125–134, 1986.

[45] J. G. Leishman and T. S. Beddoes. A generalised model for aircraft unsteady aerodynamic behaviour and dynamic stall using the indicial method. In *42nd Annual Forum of the American Helicopter Society*, Washington, DC, USA, June 1986.

[46] L. Ljung. *System Identification: Theory for the User.* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, USA, 1987.

[47] D. G. Mabey. Physical phenomena associated with unsteady transonic flows. In D. Nixon, editor, *Unsteady Transonic Aerodynamics*, volume 120 of *Progress in Astronautics and Aeronautics*, pages 1–55. AIAA, 1989.

[48] A. J. Mahajan, K. R. V. Kaza, and E. H. Dowell. Semi-empirical model for prediction of unsteady forces on an airfoil with application to flutter. *Journal of Fluids and Structures*, 7:87–103, 1993.

[49] V. Maniezzo. Genetic evolution of the topology and weight distribution of neural networks. *IEEE Transactions on Neural Networks*, 5(1):39–53, January 1994.

[50] S. F. Masri, A. G. Chassiakos, and T. K. Caugley. Identification of nonlinear dynamic systems using neural networks. *Journal of Applied Mechanics*, 60:123–133, March 1993.

[51] W. J. McCroskey. Some current research in unsteady fluid dynamics. *Journal of Fluid Engineering*, 99(1):8–39, 1977.

[52] W. J. McCroskey. Unsteady airfoils. *Annual Review of Fluid Mechanics*, 14:285–311, 1982.

[53] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs.* Springer-Verlag, 1992.

[54] M. Mitchell. *An Introduction to Genetic Algorithms.* The MIT Press, USA, 1996.

[55] D. S. Modha and R. Hecht-Nielsen. Multilayers functionals. In J. G. Taylor, editor, *Mathematical Approaches to Neural Networks*, North-Holland Mathematical Library, Amsterdam, The Netherlands, 1993. Elsevier Science Publishers B. V.

[56] B. Müller and J. Reinhart. *Neural Networks, an Introduction.* Springer-Verlag, New York, USA, 1990.

[57] K. S. Narendra and K. Parthasarathy. Identification and control of dynamical systems using neural networks. *IEEE Transactions on Neural Networks*, 1(1):4–26, March 1990.

[58] A. J. Niven and R. A. McD. Galbraith. A user guide to the Beddoes model. University of Glasgow, Report GU-AERO-9105, May 1991.

[59] R. H. J. M. Otten and L. P. P. P. van Ginneken. *The Annealing Algorithm.* Kluwer Academic Publishers, Boston, USA, 1989.

[60] T. Poggio and F. Girosi. Networks for approximation and learning. *Proceedings of the IEEE*, 78(9):1481–1497, September 1990.

[61] R. A. Rutenbar. Simulated annealing algorithms: An overview. *IEEE Circuits and Devices Magazine*, pages 19–26, January 1989.

[62] R. H. Scalan and R. Rosenbaum. *Introduction to the Study of Aircraft Vibration and Flutter.* The Macmillan Company, New York, USA, 1951.

[63] J. D. Schaffer, D. Whitley, and L. J. Eshelman. Combinations of genetic algorithms and neural networks: A survey of the state of the art. In *International Workshop on Combinations of Genetic Algorithms and Neural Networks (COGANN-92)*, pages 1–37, 1992.

[64] M. Schetzen. *The Volterra and Wiener Theories of Nonlinear Systems.* John Wiley & Sons, New York, USA, 1980.

[65] M. Schetzen. Nonlinear system modelling based on the Wiener theory. *Proceedings of the IEEE*, 69(12):1557–1573, December 1981.

[66] S. J. Schreck, W. E. Faller, and M. W. Luttges. Neural networks prediction of three-dimensional unsteady separated flowfields. *Journal of Aircraft*, 32(1):178–185, January-February 1995.

[67] W. A. Silva. Application to nonlinear systems theory to transonic unsteady aerodynamic responses. *Journal of Aircraft*, 30(5):660–668, 1993a.

[68] W. A. Silva. Extension of a nonlinear systems theory to general-frequency unsteady transonic aerodynamic responses. In *AIAA-93-1590-CP*, pages 2490–2503, 1993b.

[69] H. Tijdeman and R. Seebass. Transonic flow past oscillating airfoils. *Annual Review of Fluid Mechanics*, 12:181–222, 1980.

[70] M. Tobak and G. T. Chapman. Nonlinear problems in flight dynamics involving aerodynamic bifurcations. In *AGARD Symposium on Unsteady Aerodynamics. Fundamentals and Applications to Aircraft Dynamics*, Göttingen, West Germany, May 1985. Paper 25.

[71] M. Tobak and W. E. Pearson. A study of nonlinear longitudinal dynamic stability. NASA TR R-209, September 1964.

[72] M. Tobak and L. B. Schiff. The role of time-history effects in the formulation of the aerodynamics of aircraft dynamics. In *AGARD Conference Proceedings n. 235*, 1978.

[73] M. Tobak and L. B. Schiff. Aerodynamic mathematical modeling - basic concepts. AGARD Lecture Series n. 114, paper n. 1, 1981.

[74] C. T. Tran and D. Petot. Semi-empirical model for the dynamic stall of airfoils in view of the application to the calculation of responses of a helicopter blade in forward flight. *Vertica*, 5:35–53, 1981.

[75] E. A. Wan. Temporal backpropagation: An efficient algorithm for finite impulse response neural networks. In D. S. Touretzky, J. L. Elman, T.J. Sejnowski, and G. E. Hinton, editors, *1990 Connectionist Models Summer School*, pages 131–137, San Mateo, CA, USA, 1990a.

[76] E. A. Wan. Temporal backpropagation for FIR neural networks. In *1990 International Joint Conference on Neural Networks (IJCNN90)*, volume 1, pages 575–580, San Diego, CA, USA, 17-21 June 1990b.

[77] J. Wray and G. G. R. Green. Calculation of the Volterra kernels of non-linear dynamic systems using an artificial neural network. *Biological Cybernetics*, 71:187–195, 1994.

[78] R. R. Yager and D. P. Filev. *Essentials of Fuzzy Modeling and Control*. John Wiley & Sons, Inc., USA, 1994.