Aircraft Pitch Attitude Control
Using Backstepping

Ola Härkegård and Torkel Glad

Department of Electrical Engineering
Linköping University, SE-581 83 Linköping, Sweden
WWW: http://www.control.isy.liu.se
Email: \{ola,torkel\}@isy.liu.se

May 23, 2000

Report no.: LiTH-ISY-R-2243
Submitted to CDC 2000

Technical reports from the Automatic Control group in Linköping are available by anonymous ftp at the address ftp.control.isy.liu.se. This report is contained in the file 2243.pdf.
Aircraft Pitch Attitude Control
Using Backstepping

Ola Härkegård and Torkel Glad

May 23, 2000

Abstract

A nonlinear approach to the automatic pitch attitude control problem for a generic fighter aircraft is presented. A nonlinear model describing the longitudinal equations of motion in strict feedback form is derived. Backstepping is utilized for the construction of a globally stabilizing controller with a number of free design parameters. Two tuning schemes are proposed based on the desired locally linear controller properties. The controller is evaluated using the HIRM fighter aircraft model.

Keywords: Nonlinear control; Backstepping; Aircraft control; Local approximation.

1 Introduction

In this paper, we consider automatic pitch attitude control for a fighter aircraft. Given a pilot command on the desired pitch angle, the aircraft is automatically brought into a rectilinear motion with the nose pointing in the desired direction. This mode of operation can be useful when the pilot wants to aim on-board weapons. The control law is designed to explicitly consider the nonlinear aerodynamic forces and moments acting upon the vehicle. This is done using a backstepping design, where the control law is recursively constructed, along with a Lyapunov function guaranteeing global stability.

Backstepping, described in Section 2, relies heavily on the structure of the differential equations describing the system to be controlled. In Section 3, a nonlinear model describing the longitudinal aircraft motion in the crucial strict feedback form is derived. The control design is performed in three steps in Section 4, resulting in a state feedback control law with a number of free parameters. How to choose these, i.e., the tuning issue, is discussed in Section 5. Here, the idea is to demand a certain linear behavior locally around the reference trajectory, and two search methods are proposed for finding suitable control law parameters.

In Section 6 the backstepping control law is applied to the High Incidence Research Model (HIRM), a generic fighter aircraft model. Although the design was performed under a number of simplifying assumptions and approximations, the simulation results prove to be in good agreement with the theoretical predictions.
2 Backstepping

Backstepping is one of the successful outcomes of recent years research within constructive nonlinear control, where former descriptive tools have been “activated” to become design tools. See [8] for a design-oriented textbook and [5] for an overview of the progress during the last decade. In the following section, the basic ideas of backstepping will be presented in the setting that we will need.

Backstepping, in a quite general form, considers the system structure

\[
\begin{align*}
\dot{x} &= f(x, \xi) \\
\dot{\xi} &= u
\end{align*}
\]

where \( x \in \mathbb{R}^n, \xi \in \mathbb{R} \) are state variables and \( u \in \mathbb{R} \) is the control variable. A Lyapunov function,

\[
V_1 = \frac{1}{2} x^T L x, \quad L = L^T > 0
\]

is known for the \( \dot{x} \) subsystem such that its time derivative would satisfy

\[
\dot{V}_1 = x^T L f(x, s(x)) < 0, \quad x \neq 0
\]

if only the virtual control law

\[
\xi = s(x)
\]

could be satisfied. The key property of backstepping is that it allows us to bring (4) a “step back” through the system and find a realizable control law \( u(x, \xi) \) guaranteed to stabilize the augmented system, (1). First, introduce the residual

\[
z = \xi - s(x)
\]

and rewrite (1) in terms of \( z \).

\[
\begin{align*}
\dot{x} &= f(x, s(x) + z) = f(x, s(x)) + \psi(x, z)z \\
\dot{z} &= u - \dot{s}(x) = u - \frac{\partial s(x)}{\partial x}(f(x, s(x)) + \psi(x, z)z)
\end{align*}
\]

Here we have introduced

\[
\psi(x, z) = \frac{f(x, s(x) + z) - f(x, s(x))}{z}
\]

where the limit derivative, as \( z \to 0 \), is required to exist. Now expand the Lyapunov function into

\[
V_2 = V_1 + \frac{1}{2} lz^2, \quad l > 0
\]

with the second term penalizing any deviation from the virtual control law. Finally, compute its time derivative and demand a stabilizing behavior.

\[
\begin{align*}
\dot{V}_2 &= x^T L(f(x, s(x)) + \psi(x, z)z) + lz(u - \dot{s}(x)) \\
&= \dot{V}_1|_{z=0} + (x^T L\psi(x, z) + l(u - \dot{s}(x)))z \\
&= \dot{V}_1|_{z=0} - \lambda z^2
\end{align*}
\]
Solving for $u$ yields

$$u = s(x) - \frac{1}{l}x^T L \psi(x, z) + \lambda z$$

$$= \frac{\partial s(x)}{\partial x} (f(x, s(x)) + \psi(x, z)z) - \frac{1}{l}x^T L \psi(x, z) + \lambda z$$

(10)

A natural extension is to allow for $u$ to be state dependent through a mapping

$$u = g(x, \xi, u')$$

(11)

where $u'$ is the actual control variable. Then the criterion for (10) to be propagated into a control law, $u'(x, \xi)$, is that $g$ must be invertible with respect to $u'$.

One class of systems, to which the backstepping technique can be successfully applied, is strict feedback form systems, described by

$$\dot{x}_1 = f_1(x_1, x_2)$$

$$\vdots$$

$$\dot{x}_{n-1} = f_{n-1}(x_1, \ldots, x_{n-1}, x_n)$$

$$\dot{x}_n = f_n(x_1, \ldots, x_n, u)$$

(12)

where $f_2, \ldots, f_n$ are assumed to be invertible with respect to their last argument. The backstepping design procedure is straightforward. First, a virtual control law $x_2 = s_1(x_1)$ is constructed guaranteeing global stability of the $\dot{x}_1$ subsystem. Through steps (5)-(10), this propagates to the next state, resulting in a new virtual control law $x_3 = s_2(x_1, x_2)$, which, if it were realizable, would achieve global stability of the $(\dot{x}_1, \dot{x}_2)$ subsystem. Eventually, this recursive approach leads to a state feedback control law, $u(x_1, \ldots, x_n)$, assuring global stability of the complete system.

3 Nonlinear aircraft model

The longitudinal motion of an aircraft is well described by the following standard set of differential equations, see [1]. The state variables, $V$ (air speed), $\alpha$ (angle of attack), $\theta$ (pitch angle), and $q$ (pitch rate), are depicted in Figure 1.

$$\dot{V} = \frac{1}{m}(-D + F_T \cos \alpha - mg \sin \gamma)$$

(13a)

$$\dot{\alpha} = \frac{1}{mV}(-L - F_T \sin \alpha + mg \cos \gamma) + q$$

(13b)

$$\dot{\theta} = q$$

(13c)

$$\dot{q} = \frac{1}{I_y} (M + F_T Z_{TP})$$

(13d)

$\gamma = \theta - \alpha$ is the flight path angle, to which we shall return later. $F_T$ is the engine thrust force which contributes to the pitching moment due to the thrust point offset, $Z_{TP}$. The drag force, $D$, the lift force, $L$, and the pitching moment,
\( D = \bar{q}SC_D(\alpha, \delta), \quad L = \bar{q}SC_L(\alpha, \delta), \quad M = \bar{q}SC_m(\alpha, q, \delta) \) (14)

Here we have introduced the aerodynamic pressure, \( \bar{q} = \frac{1}{2} \rho V^2 \), \( \rho \) being the air density, the wing platform area, \( S \), and the mean aerodynamic chord, \( \bar{c} \). We let \( \delta \) represent the angles of the control surfaces that are at our disposal.

Comparing with (12), (13) is obviously not in the strict feedback form that is necessary for backstepping. The following modifications and approximations are introduced:

- Neglect the variation in air speed, i.e., use \( \dot{V} = 0 \).
- Trade \( \alpha \) for \( \gamma \) as a state variable, thereby removing the \( q \) dependence in (13b).
- Use the fact that the deflection of the control surfaces mainly acts on the aircraft through the pitching moment and not so much through the lift force (see [3]), i.e., \( C_L \approx C_L(\alpha) \). This removes the \( \delta \) dependence in (13b).

The resulting revised system description is now in strict feedback form.

\[
\begin{align*}
\dot{\gamma} &= \frac{1}{mV}(\bar{q}SC_L(\alpha) + F_T \sin \alpha - mg \cos \gamma) \quad (15a) \\
\dot{\theta} &= q \\
\dot{q} &= \frac{1}{I_y}(\bar{q}SC_m(\alpha, q, \delta) + F_T Z_{TP}) \quad (15c)
\end{align*}
\]

4 Constructing the backstepping controller

The aim of the controller is to bring the aircraft into rectilinear flight with the direction commanded by the pilot. It may be that the nose should point
in a certain direction, $\theta_{\text{ref}}$, it may also be that the pilot commands a certain flight path angle, $\gamma_{\text{ref}}$. Irrespective of the situation, the desired steady state is described by

$$\gamma = \gamma_{\text{ref}}, \quad \theta = \theta_{\text{ref}}, \quad \alpha = \alpha_{\text{ref}} = \theta_{\text{ref}} - \gamma_{\text{ref}}, \quad q = 0 \quad (16)$$

Let us outline the construction of the controller to performed in three steps.

1. Consider (15a) and use physical understanding to find a virtual control law, $\theta = s_1(\gamma)$, that would stabilize $\gamma$ around $\gamma_{\text{ref}}$.

2. Use backstepping, as described in Section 2, to propagate $s_1$ two steps back through the system (15) to find the required $\dot{q} = f_q$ that will bring the complete system to the desired steady state.

3. Find, if possible, deflections $\delta$ that will produce this $f_q$.

**First step** Introduce the residual $z_1 = \gamma - \gamma_{\text{ref}}$ and rewrite (15a) as

$$\dot{z}_1 = f_{z_1}(z_1, \theta) = \frac{1}{mV}(\dot{q}SC_L(\alpha) + F_T \sin \alpha - mg \cos(z_1 + \gamma_{\text{ref}})) \quad (17)$$

The typical characteristics of the lift coefficient, $C_L$, are illustrated in Figure 2, using data from the test vehicle in Section 6. Neglecting the gravitational term of $f_{z_1}$, and using the sign properties of the two other terms, we see that

$$\text{sign } f_{z_1} = \text{sign } (\alpha - \alpha_{\text{ref}}) \quad (18)$$
for a wide range of $\alpha$ since $f_{z_1} = 0$ at the equilibrium. This is the key to the
construction of our opening virtual control law. Simply aim at giving the virtual
control law

$$\alpha - \alpha_{ref} = -\lambda_1 z_1, \quad \lambda_1 > 0 \quad \Leftrightarrow$$

$$\theta = s_1(z_1) = (1 - \lambda_1)z_1 + \theta_{ref}$$

(19)

We can confirm the virtual stability of (15a) using the Lyapunov function

$$V_1 = \frac{1}{2} z_1^2$$

(20)

satisfying

$$\dot{V}_1|_{\theta = s_1(z_1)} = z_1 f_{z_1}(z_1, s_1(z_1)) = -W_1(z_1; \lambda_1)$$

$$\approx z_1 \frac{1}{mL} q \bar{F} L (-\lambda_1 z_1 + \alpha_{ref}) < 0, \quad z_1 \neq 0$$

(21)

Second step Propagating (19) one step through the system, to get the next
virtual control law, is straightforward. Our system consists of (17) augmented
by (15b). Applying steps (5)-(10) yields

$$q = s_2(z_1, z_2) = \dot{s}_1 - \frac{1}{l_2}(z_1 \psi_1(z_1, z_2) + \lambda_2 z_2)$$

$$= (1 - \lambda_1) f_{z_1}(z_1, \theta) - \frac{1}{l_2}(z_1 \psi_1(z_1, z_2) + \lambda_2 z_2)$$

(22)

where

$$z_2 = \theta - s_1(z_1)$$

$$\psi_1(z_1, z_2) = \frac{f_{z_1}(z_1, s_1(z_1) + z_2) - f_{z_1}(z_1, s_1(z_1))}{z_2}$$

(23)

(24)

The virtual stability is shown by extending (20) with a $z_2$ term. The construc-
tion of $s_2$ guarantees that

$$V_2 = V_1 + \frac{1}{2} l_2 z_2^2$$

(25)

satisfies

$$\dot{V}_2|_{q = s_2(z_1, z_2)} = \dot{V}_1|_{z_2 = 0} - \lambda_2 z_2^2$$

(26)

The final iteration of our backstepping design gives us

$$f_q = \dot{s}_2 - \frac{1}{l_3}(l_2 z_2 \psi_2 + \lambda_3 z_3)$$

(27)

where

$$z_3 = q - s_2(z_1, z_2)$$

(28)

and $\psi_2 = 1$ (due to the simplicity of (15b)). Computing $f_q$ apparently involves
computing the time derivative of $s_2$, and we refer to Appendix A for the details.

If deflections $\delta$ can be found such that (27) is satisfied, the stability of the
complete system (15) is now described by

$$V_3 = V_2 + \frac{1}{2} l_3 z_3^2 = \frac{1}{2}(z_1^2 + l_2 z_2^2 + l_3 z_3^2)$$

$$\dot{V}_3 = \dot{V}_2|_{z_3 = 0} - \lambda_3 z_3^2 = -W_1(z_1; \lambda_1) - \lambda_2 z_2^2 - \lambda_3 z_3^2$$

(29)
Third step The last step of the controller design consists of finding proper control surface deflections to satisfy (27), i.e., to solve
\[ \frac{1}{I_y}(\ddot{q}S\bar{c}C_m(\alpha, q, \delta) + F_TZ_{TP}) = f_q \] (30)
for \( \delta \). This task is twofold. One part involves the moment \( F_TZ_{TP} \) produced by the engine. This quantity can not be measured and has to be estimated. The other is to invert the pitching coefficient, \( C_m \), with respect to \( \delta \). We will now deal with these issues.

A pragmatic approach for estimating \( F_T \), or rather its contribution to \( \dot{q} \), is to model (15c) as
\[ \dot{q} = \frac{1}{I_y}q\ddot{\bar{c}}C_m(\alpha, q, \delta) + E \]
\[ \dot{E} = w \] (31)
where \( w \) is white noise with a properly selected variance. Using measurements of \( q, E \) can be successfully tracked as the thrust force changes over time, using a simple Kalman filter.

With \( E \) at our disposal, we can solve (30) for \( C_m \).
\[ C_m(\alpha, q, \delta) = \frac{I_y}{q\ddot{\bar{c}}}(f_q - E) \] (32)

The HIRM configuration offers two ways of controlling the pitching coefficient; via the taileron at the back of the aircraft, and via the canard wings. A number of ways of how to divide the work between these control surfaces have been suggested. In this paper, however, we put no effort into optimizing this aspect but simply ignore the canard wings, leaving everything to the taileron. In the following, \( \delta \) will therefore represent the taileron's angle of deflection, see Figure 1.

The pitching coefficient, \( C_m \), is usually a complicated function of the arguments involved. The standard way of modeling it is to collect measurements for a number of different situations and create a look-up table from which intermediate values can be interpolated. Assuming such a table is available to us, it can be used to solve (32) for \( \delta \). Given measurements of \( \alpha \) and \( q \), the left hand side typically becomes a close to monotonic function of \( \delta \). This fits many numerical solvers well, and the Van Wijngaarden-Dekker-Brent method, see [2] and [4], was chosen for solving (32). This method is a happy marriage between bisection, providing sureness of convergence, and inverse quadratic interpolation, providing superlinear convergence in the best-case scenario.

Naturally, there are hard bounds on the possible taileron deflections. For the HIRM, \( \delta \in [-40^\circ, 10^\circ] \) is the possible range. In cases where (32) cannot be satisfied, i.e., when enough pitching moment cannot be produced, we simply saturate and choose the proper \( \delta \) bound.

We can now, at least formally, write the resulting state feedback backstepping control law as
\[ \delta = \delta_{bs}(\gamma, \theta, q) \] (33)
5 Tuning the controller

The resulting controller (33) is parameterized by the five parameters $l_2, l_3, \lambda_1, \lambda_2,$ and $\lambda_3$. Equation (29) ensures stability for any choice $l_i > 0, \lambda_i > 0$. What is then a suitable choice? Intuition gives us little insight in how to manually adjust the parameters to achieve a certain desired closed loop behavior. A natural thing to do is to ask which linear controller (33) approximates locally for certain choice of parameters. The reverse question is even more interesting.

Given a linear controller,

$$\delta_{lin} = \begin{pmatrix} k_1 & k_2 & k_3 \end{pmatrix} \begin{pmatrix} \gamma - \gamma_{ref} \\ \theta - \theta_{ref} \\ q \end{pmatrix} = Kx \quad (34)$$

can we find $l_i, \lambda_i$ such that (33) locally achieves (34)? We will not answer the question for which choices of $K$ this is possible but merely propose two search schemes.

5.1 A linearization approach

The obvious way to answer the question posed is to linearize (33) and solve the system of equations to determine $l_i, \lambda_i$ from $K$. Even for our relatively small system though, this turns out to be a tedious task to do by hand. The computer based alternative is to use a nonlinear root finding scheme to solve

$$\nabla \delta bs(x)|_{x=0} = K \quad (35)$$

where the gradient is computed numerically given $l_i, \lambda_i$.

5.2 A Lyapunov approach

A different approach is to consider the desired Lyapunov stability, defined by (29). Assume that the system (15) locally around the level flight equilibrium (where $\gamma_{ref} = 0$), can be described by

$$\dot{x} = Ax + B\delta \quad (36)$$

Given (34), the desired closed loop behavior is then

$$\dot{x} = (A + BK)x = Ax \quad (37)$$

To express this in terms of $z = \begin{pmatrix} z_1 & z_2 & z_3 \end{pmatrix}$, we need the mapping from $x$ to $z$. Locally,

$$z = T_{zx}x = \nabla Z(x)|_{x=0} x \quad (38)$$

where $Z(x)$ is the nonlinear mapping defined by (23) and (28). We have that

$$T_{zx} = \left. \begin{pmatrix} 1 & 0 & 0 \\ -\frac{\partial s_1}{\partial \gamma} & 1 & 0 \\ -\frac{\partial s_2}{\partial \gamma} & -\frac{\partial s_2}{\partial \theta} & 1 \end{pmatrix} \right|_{x=0} \quad (39)$$
In terms of \( z \), (37) becomes
\[
\dot{z} = T_{zx} A_z T_{zz}^{-1} z = A_z z \quad (40)
\]

We now have the setup ready to return to the desired Lyapunov stability, (29). We can write this as
\[
\frac{d}{dt} \left[ \frac{1}{2} z^T \begin{pmatrix} 1 & 0 & 0 \\ 0 & l_2 & 0 \\ 0 & 0 & l_3 \end{pmatrix} z \right] = -z^T \begin{pmatrix} \lambda_1' & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix} z \quad (41)
\]

where
\[
\lambda_1' = \frac{1}{2} \frac{d^2 W_1}{d z_1^2} (0; \lambda_1) = -\frac{df_{z_1}}{dz_1} (0, s_1(0)) \quad (42)
\]

which gives the matrix equation
\[
A_z^T L + L A_z + 2\Lambda = 0 \quad (43)
\]

to be solved for \( L \) and \( \Lambda \). Note that \( A_z \) also depends on the design parameters through the transformation matrix, \( T_{zx} \). A numerically sound way of computing \( L, \Lambda \) is to settle for parameters that minimize the left hand side in some sense, e.g., using the 2-norm or the Frobenius norm.

The difference in this approach to the brute force linearization approach is that \( s_3 \) does not appear explicitly but only the easily computed derivatives of \( s_1 \) and \( s_2 \) in (39).

6 Application

In this section we investigate the properties of the derived control law, (33), applied to High Incidence Research Model (HIRM), originally released for the GARTEUR robust flight control challenge, documented in [7]. HIRM is a realistic model of generic fighter aircraft, including, e.g., actuator and sensor dynamics.

For the simulations below, a flight case where the aircraft is in level flight at Mach 0.3 and at a height of 5000 ft was chosen. To maintain the initial speed, a simple speed controller was included, borrowed from [6]. The pitch controller was tuned to locally achieve \( K = (0.3150 \quad 0.4024 \quad 0.5337) \), resulting from an LQ-design. (43) was used to find nearly optimal values of \( l_i, \lambda_i \) which in turn were fine-tuned using (35). The resulting parameters were
\[
l_2 = 3.9131 \quad \lambda_1 = 0.9013
\]
\[
l_3 = 2.3707 \quad \lambda_2 = 3.1243 \quad \lambda_3 = 2.3583 \quad (44)
\]

Figure 3 shows the closed loop behavior when the reference trajectory is two consecutive steps in the pitch angle, \( \theta \). It is interesting at this point to see how well our controller performs on the full system, including variations in speed and sensor and actuator dynamics. These effects were not regarded in the
control design but rely on the robustness properties of the controller. A simple, visual check is achieved from Figure 4. Here, the designed Lyapunov time derivative from (29), $\dot{V}_3$, has been plotted against the estimated time derivative of the Lyapunov function. As we can see, the fit is bad right after a step in the reference signal, due to actuator dynamics, whereafter the curves converge nicely.

7 Conclusions

We have addressed the pitch attitude control problem for a generic fighter aircraft. A nonlinear model describing the longitudinal motion in strict feedback format was derived, carefully selecting the states included and making proper approximations. A backstepping control law was designed recursively in three steps, resulting in a nonlinear controller with five free parameters. For tuning the controller a brute force linearization approach was suggested along with a method based on the desired Lyapunov stability of the closed loop system. The idea behind both of these was to locally achieve a given linear controller. Computer simulations using the HIRM model proved the controller to work as expected and to be robust against the approximations made in the controller design.
Figure 4: Lyapunov stability. Above: The Lyapunov function, $V_3$, from (29). Below: The designed time derivative from (29), $\dot{V}_3$ (dashed), along with the estimated actual time derivative (solid).
8 Acknowledgments

The authors would like to acknowledge Karin Ståhl-Gunnarsson at SAAB AB for her insightful comments and valuable expertise during the course of this work.

References


A Derivation details

In this appendix, we return to the desired pitch rate, $f_q$, of Equation (27), and explicitly calculate the components of the $\dot{s}_2$ term.

Recall from Equation (22) that

$$s_2(z_1, z_2) = (1 - \lambda_1)f_{z_1}(z_1, \theta) - \frac{1}{l_2}(z_1\psi_1(z_1, z_2) + \lambda_2 z_2)$$

Differentiating with respect to time yields

$$\dot{s}_2 = \frac{\partial s_2}{\partial z_1} \dot{z}_1 + \frac{\partial s_1}{\partial z_2} \dot{z}_2$$

Below, the building bricks of this expression are scrutinized and expanded until into directly computable entities.

$$\dot{z}_1 = f_{z_1}(z_1, \theta)$$
$$\dot{z}_2 = q - \dot{s}_1 = q - (1 - \lambda_1)f_{z_1}(z_1, \theta)$$

$$\frac{\partial s_2}{\partial z_1} = (1 - \lambda_1)(\frac{\partial f_{z_1}}{\partial z_1} + \frac{\partial f_{z_1}}{\partial \theta} \frac{\partial \theta}{\partial z_1}) - \frac{1}{l_2}(\psi_1 + z_1 \frac{\partial \psi_1}{\partial z_1})$$

$$\frac{\partial s_2}{\partial z_2} = (1 - \lambda_1)(\frac{\partial f_{z_1}}{\partial \theta} \frac{\partial \theta}{\partial z_2}) - \frac{1}{l_2}(z_1 \frac{\partial \psi_1}{\partial z_2} + \lambda_2)$$

$$\frac{\partial f_{z_1}}{\partial z_1} = \frac{1}{mV}(\bar{qS} \frac{dC_L}{d\alpha} \frac{\partial \alpha}{\partial z_1} - F_T \cos \alpha \frac{\partial \alpha}{\partial z_1} + mg \sin(z_1 + \gamma_{ref}))$$

$$\frac{\partial f_{z_1}}{\partial \theta} = \frac{1}{mV}(\bar{qS} \frac{dC_L}{d\alpha} \frac{\partial \alpha}{\partial \theta} + F_T \cos \alpha \frac{\partial \alpha}{\partial \theta})$$

$$\frac{\partial \psi_1}{\partial z_1} = \frac{1}{z_2}(\frac{\partial f_{z_1}}{\partial z_1} + \frac{\partial f_{z_1}}{\partial \theta} \frac{\partial \theta}{\partial z_1}) - \frac{1}{l_2}(z_1, \lambda_1) \frac{\partial f_{z_1}}{\partial z_1}(z_1, \lambda_1) + \frac{\partial f_{z_1}}{\partial z_1}(z_1, s_1) \frac{\partial s_1}{\partial z_1(1 - \lambda_1)}$$

$$\frac{\partial \psi_1}{\partial z_2} = \frac{1}{z_2}(\frac{\partial f_{z_1}}{\partial \theta} \frac{\partial \theta}{\partial z_2}) - \frac{1}{l_2} \frac{\partial f_{z_1}}{\partial z_2}$$

The bottom line is that what we need, in terms of aerodynamical data, to compute $f_q$, is the value and the gradient of $L$ at the actual and at the desired angle of attack..