Chapter 6. Level Flight Control

In level flight a tailsitter acts as a normal fixed-wing aircraft. Because of this, standard level flight control techniques can easily be applied to a tailsitter. In most control schemes the longitudinal, altitude and airspeed, and lateral, position and course, are assumed to be decoupled and are controlled separately. We will follow this tradition and have separate longitudinal and lateral control schemes. In this chapter we assume that we do not have a high fidelity aerodynamic model of the aircraft and that low cost sensors are used.

A common approach to control the airspeed and altitude for a fixed-wing aircraft is to assume that the airspeed and altitude dynamics are decoupled. Using this assumption, the altitude is controlled by the elevator while the thrust controls the airspeed [16]. While this type of controller has been used successfully on many autopilots, the underlying assumption that the airspeed and altitude dynamics are decoupled is inherently false. For example, consider the simple case when an aircraft pitches up while the thrust does not change. If the dynamics were truly decoupled, the aircraft's altitude would increase while the airspeed would remain unchanged. However, as should be obvious, the airspeed will decrease while the altitude increases. In other words, some of the aircraft's kinetic energy is converted to potential energy.

In the early 1980's, Lambregts et al. realized that the airspeed and altitude of an aircraft could be controlled by manipulating the kinetic and potential energy of the system [65–67]. Approaching the problem this way allowed the coupling between the altitude and airspeed to be taken into account. Controllers based on this idea, called the Total Energy Control System (TECS), have been successfully used and tested on a variety of airframes [20,21,114]. While most of these controllers are of PI type, other variants have been developed such as adding pitch damping [113]. Viswanathan et al. use L1 adaptive control to make the

energy rates follow a reference model which eliminates the need for gain scheduling [112]. TECS ideas have also been explored to see if cockpit displays could be developed to present pilots with this information [8] and how they could be used as instructional tools [77]. TECS concepts have also been applied to the lateral control of an aircraft [22, 47], for the longitudinal control of a helicopter [29], and the translational control of a quadcopter [111].

In 2010, Akmeliawati and Mareels derived a nonlinear energy based control method for the longitudinal control of a fixed-wing airplane that has some similarities to the TECS [7]. In their approach they decompose the dynamics into fast dynamics, which account for the pitch and pitch rate, and slow dynamics, which account for everything else. Elevator and throttle controllers are then derived which provide stability and tracking guarantees. However, their controller requires accurate model parameters and model inaccuracies of only $\pm 2\%$ have been tested.

In this chapter, we derive a nonlinear TECS altitude and airspeed controller that is easy to apply to different airframes. Two variants of this controller are compared, through simulation, to the original TECS controller, several other TECS type controllers, and a standard decoupled successive loop closure controller. These simulations show that the nonlinear TECS controllers account for the coupling between the airspeed and altitude dynamics while having smaller oscillations in the response as compared to the original TECS control scheme. In addition, the nonlinear controllers have a better response than the other controllers.

This chapter is organized in the following manner. Section 6.1 defines the various energy values and their derivatives. The original TECS control scheme is described in Section 6.2. Section 6.3 contains the motivation and derivation for the nonlinear TECS controller, and Section 6.4 describes simple pitch and thrust control schemes that can be used while implementing the nonlinear TECS controller in hardware. For comparison, three other controllers are briefly described in Section 6.5. The simplified three degree of freedom dynamic model is described in Section 6.7, a simple parameter estimation scheme is presented in Section 6.8, and simulation results are shown in Section 6.9. Finally, conclusions are presented in Section 6.10.

6.1 Energy Definitions

If the aircraft is modeled as a point mass then the total energy is the kinetic energy plus the potential energy,

$$E_T \triangleq \mathrm{m}gh + \frac{1}{2}\mathrm{m}V_a^2, \tag{6.1}$$

and the total energy rate is

$$\dot{E}_T = \mathrm{m}g\dot{h} + \mathrm{m}V_a\dot{V}_a. \tag{6.2}$$

The energy difference is defined to be the potential energy minus the kinetic energy,

$$E_D \triangleq \mathrm{m}gh - \frac{1}{2}\mathrm{m}V_a^2, \tag{6.3}$$

and the energy distribution rate, or the energy difference rate, is

$$\dot{E}_D = \mathrm{m}g\dot{h} - \frac{1}{2}\mathrm{m}V_a\dot{V}_a. \tag{6.4}$$

The airspeed, acceleration, altitude, and climb rate are easily computed from the energy definitions and are

$$\begin{aligned} V_a &= \left(\frac{E_T - E_D}{\mathrm{m}}\right)^{\frac{1}{2}} \\ \dot{V}_a &= \left(\frac{\mathrm{m}}{E_T - E_D}\right)^{\frac{1}{2}}\left(\dot{E}_T - \dot{E}_D\right) \\ h &= \frac{E_D + E_T}{2\mathrm{m}g} \\ \dot{h} &= V_a \sin(\gamma) = \frac{\dot{E}_D + \dot{E}_T}{2\mathrm{m}g}. \end{aligned} \tag{6.5}$$

Typically, TECS controllers use a scaled version of the total energy rate and energy distribution [35]. Rearranging (6.2) provides

$$\dot{E}_t = \frac{\dot{E}_T}{\mathrm{m}gV_a} = \frac{\dot{V}_a}{g} + \frac{\dot{h}}{V_a}. \tag{6.6}$$

If we assume that γ is small then (6.6) becomes

$$\dot{E}_t = \frac{\dot{V}_a}{g} + \gamma$$

which is the standard total energy rate value used in most TECS control schemes. Likewise, the energy distribution rate, (6.4), can be rearranged to

$$\dot{E}_d = \frac{\dot{E}_D}{\mathrm{m}gV_a} = -\frac{\dot{V}_a}{g} + \gamma.$$

6.2 Original TECS

Every TECS altitude and airspeed controller shares the same basic principles. First, energy is conserved. Kinetic energy, or the airplane's airspeed, can be converted into potential energy, or altitude, and vice versa. Second, assuming a point mass model of the aircraft, the thrust generated by the propulsion system is the only way to add energy to the system and drag is the only way that energy is removed. TECS controllers also assume that the angle of attack is low and the flight path angle does not influence drag. Using these basic principles and assumptions the TECS longitudinal controller can be developed.

As mentioned earlier, the total energy rate is controlled by the thrust which implies that the commanded thrust should be a function of $\dot{E}_t$. Likewise, the elevator is approximately energy conservative and allows kinetic energy and potential energy to be converted to each other which implies that the elevator should be used to control the energy distribution rate. Using these ideas, the thrust command is

$$T^c = T_D + \Delta T,$$

where T_D is the trim thrust needed to counteract drag and

$$\Delta T = k_{p,T}\dot{E}_t + k_{i,T}\int_{t_0}^{t}\Big(\dot{E}_t^d - \dot{E}_t\Big)\delta\tau, \tag{6.7}$$

where $k_{i,T}$ and $k_{p,T}$ are the integral and proportional gains respectively and $\dot{E}_t^c$ is the commanded total energy rate computed from the commanded acceleration and flight path angle. Likewise, the pitch command is

$$\theta^c = k_{p,\theta}\dot{E}_d + k_{i,\theta}\int_{t_0}^{t}\Big(\dot{E}_d^d - \dot{E}_d\Big)\delta\tau, \tag{6.8}$$

where $k_{i,\theta}$ and $k_{p,\theta}$ are the integral and proportional gains respectively and $\dot{E}_d^c$ is the commanded energy distribution rate. This control scheme assumes that there are fast, low level control loops that control T and θ. The desired acceleration and flight path angle are created by a proportional term [66]

$$\dot{h}^d = k_h(h^c - h) \tag{6.9}$$

$$\gamma^d = \frac{\dot{h}^d}{V_a}$$

$$\dot{V}_a^d = k_v(V_a^c - V_a), \tag{6.10}$$

where $k_h > 0$ and $k_v > 0$ are scaling parameters. In other words the desired acceleration (climb rate) is the scaled airspeed (altitude) error.

Over the years, variations of the controller given by (6.7) and (6.8) have been developed. For example, a pitch damping term, $-k_{d,\theta}q$, is added to (6.8) in [113]. In addition there have been several different approaches to tune the gains. Faleiro and Lambregts used eigenstructure assignment to systematically select the gains in [35] and an optimization algorithm with a cost function that included the mean square of the control response needed to reject Dryden turbulence and command tracking performance, among others, selected the gains in [113]. In addition, the preferred structure of the controller was changed to improve performance. The thrust controller does not change but the pitch control scheme is modified to [68]

$$\theta^c = k_{p,\theta}\left((2-k)\gamma - k\frac{\dot{V}_a}{g}\right) + k_{i,\theta}\int_{t_0}^{t}\Big(\dot{E}_d^d - \dot{E}_d\Big)\delta\tau. \tag{6.11}$$

In this method, the true energy distribution rate is not used. Rather a weighted difference between the potential and kinetic energy rates is used. The weighting is determined by

$k \in [0, 2]$, where $k = 0$ means only the scaled potential energy rate is used, $k = 2$ means only the scaled kinetic energy rate is used, and $k = 1$ means the true energy distribution rate is used.

6.3 Nonlinear TECS

The TECS concept can be used to derive an adaptive nonlinear control scheme which attempts to control E_T and E_D directly. As mentioned, the total energy can only change due to the thrust or drag on the aircraft otherwise the energy is just transitioning between kinetic and potential. This means that the total energy rate can be derived by looking at the case where the velocity changes and the altitude is constant. Assuming $\dot{h} = 0$, the total energy rate is

$$\dot{E}_T = \mathrm{m} V_a \dot{V}_a = V_a F,$$

where F is the net non-conservative force on the aircraft. Assuming that the angle of attack is low and that the thrust and drag are aligned then

$$\dot{E}_T = V_a (T - D). \tag{6.12}$$

In general, when $\dot{h} \neq 0$,

$$\mathrm{m} g \dot{h} + \mathrm{m} V_a \dot{V}_a = \dot{E}_T = V_a (T - D)$$

which means

$$\dot{E}_D = 2\mathrm{m} g \dot{h} - V_a (T - D). \tag{6.13}$$

Because the drag is unknown we model it as

$$D = \hat{D} + \boldsymbol{\phi}^\top(\boldsymbol{x}) \boldsymbol{\Psi},$$

where $\hat{D}$ is the aerodynamic model's estimate of drag, $\boldsymbol{\phi}(\boldsymbol{x}) \in \Re^n$ is a vector of known bounded basis functions, and $\boldsymbol{\Psi} \in \Re^n$ is a vector of unknown parameters.

Let the Lyapunov function be

$$V = \frac{1}{2}\Gamma_T \tilde{E}_T^2 + \frac{1}{2}\Gamma_D \tilde{E}_D^2 + \tilde{\mathbf{\Psi}}^\top \tilde{\mathbf{\Psi}}, \tag{6.14}$$

where $\tilde{E} \triangleq E^d - E$ is the error, $\Gamma_T > 0$, and $\Gamma_D > 0$. Taking the derivative of (6.14) and using (6.5), (6.12), and (6.13) gives

$$\begin{aligned}\dot{V} = {} & \Gamma_T \tilde{E}_T \Big(\dot{E}_T^d - \Big(T - \hat{D} - \boldsymbol{\phi}^\top \mathbf{\Psi} \Big) V_a \Big) \\ & + \Gamma_D \tilde{E}_D \Big(\dot{E}_D^d - 2\mathrm{m} g V_a \sin\gamma + \Big(T - \hat{D} - \boldsymbol{\phi}^\top \mathbf{\Psi} \Big) V_a \Big) \\ & + \tilde{\mathbf{\Psi}}^\top \dot{\hat{\mathbf{\Psi}}},\end{aligned} \tag{6.15}$$

where we assume that the unknown parameters are slowly varying. If the thrust is chosen to be

$$T^c = \hat{D} + \boldsymbol{\phi}^\top \hat{\mathbf{\Psi}} + \frac{\dot{E}_T^d}{V_a} + k_T \frac{\tilde{E}_T}{V_a}, \tag{6.16}$$

where $k_T > 0$, then (6.15) becomes

$$\begin{aligned}\dot{V} = {} & -k_T \Gamma_T \tilde{E}_T^2 \\ & + \Gamma_D \tilde{E}_D \Big(\dot{E}_D^d - 2\mathrm{m} g V_a \sin\gamma + \dot{E}_T^d + k_T \tilde{E}_T \Big) \\ & + \tilde{\mathbf{\Psi}}^\top \Big(\dot{\hat{\mathbf{\Psi}}} + \Big(-\Gamma_T \tilde{E}_T + \Gamma_D \tilde{E}_D \Big) \boldsymbol{\phi} V_a \Big).\end{aligned} \tag{6.17}$$

If the flight path angle is chosen to be

$$\gamma^c = \sin^{-1}\left(\frac{\dot{E}_D^d + \dot{E}_T^d + k_T \tilde{E}_T + k_D \tilde{E}_D}{2\mathrm{m} g V_a} \right), \tag{6.18}$$

where $k_D > 0$, then (6.17) becomes

$$\begin{aligned}\dot{V} = {} & -k_T \Gamma_T \tilde{E}_T^2 - k_D \Gamma_D \tilde{E}_D^2 \\ & + \tilde{\mathbf{\Psi}}^\top \Big(\dot{\hat{\mathbf{\Psi}}} + \Big(-\Gamma_T \tilde{E}_T + \Gamma_D \tilde{E}_D \Big) \boldsymbol{\phi} V_a \Big).\end{aligned} \tag{6.19}$$

Note that (6.18) can be simplified, by using (6.5), to

$$\gamma^c = \sin^{-1}\left(\frac{\dot{h^d}}{V_a} + \frac{1}{2\mathrm{m}gV_a}\Big(k_T\tilde{E}_T + k_D\tilde{E}_D\Big)\right) \tag{6.20}$$

and that the commanded flight path angle can be converted to a pitch command by $\theta^c = \gamma^c + \alpha$.

The parameter adaption rate is chosen to be

$$\dot{\tilde{\mathbf{\Psi}}} = \Big(\Gamma_T\tilde{E}_T - \Gamma_D\tilde{E}_D\Big)\boldsymbol{\phi}V_a \tag{6.21}$$

so (6.19) becomes

$$\dot{V} = -k_T\Gamma_T\tilde{E}_T^2 - k_D\Gamma_D\tilde{E}_D^2 \tag{6.22}$$

which is negative semi-definite. By realizing that V is lower bounded and assuming that the desired altitude and airspeed are bounded, we know that $V(t) \leq V(t_0)\ \forall\ t \geq t_0$. This implies that $\tilde{E}_T$, $\tilde{E}_D$, and $\tilde{\mathbf{\Psi}}$ are bounded. The derivative of (6.22) is

$$\ddot{V} = -2k_T\Gamma_T\tilde{E}_T\dot{\tilde{E}}_T - 2k_D\Gamma_D\tilde{E}_D\dot{\tilde{E}}_D. \tag{6.23}$$

Inserting the error rates and the commanded thrust and flight path angle into (6.23) provides

$$\begin{aligned}\ddot{V} = &-2k_T^2\Gamma_T\tilde{E}_T^2 - 2k_D^2\Gamma_D\tilde{E}_D^2 \\ &+ 2\tilde{\mathbf{\Psi}}^\top\Big(k_T\Gamma_T\tilde{E}_T - k_D\Gamma_D\tilde{E}_D\Big)\boldsymbol{\phi}V_a\end{aligned}$$

which is bounded and finite. From this we can conclude, using Barbalat's Lemma [55], that $\dot{V} \to 0$ which implies $\tilde{E}_T \to 0$ and $\tilde{E}_D \to 0$.

There are several interesting things about the nonlinear TECS controller given by (6.16) and (6.20). First, if we assume that the drag is known and the thrust has been

trimmed to counteract drag then the commanded change in thrust is

$$\Delta T = \frac{\dot{E}_T^d}{V_a} + k_T \frac{\tilde{E}_T}{V_a}$$

which can be thought of as a PI controller where the proportional gain is 1 and the integral gain is k_T. This is very similar to the original TECS controller (6.7). However, this derivation suggests that the desired total energy should be used instead of the actual total energy.

Second, using the energy definitions (6.1) and (6.3) (6.20) becomes

$$\sin(\gamma^c) = \frac{\dot{h^d}}{V_a} + \frac{1}{2\mathrm{m}gV_a}\Big((k_T - k_D)\tilde{E}_K + (k_T + k_D)\tilde{E}_P\Big),$$

where

$$\tilde{E}_K \triangleq \frac{1}{2}\mathrm{m}\Big(\big(V_a^d\big)^2 - V_a^2\Big)$$
$$\tilde{E}_P \triangleq \mathrm{m}g\big(h^d - h\big).$$

This controller has three distinct behaviors depending on the relative magnitude of k_T and k_D. If $0 < k_D < k_T$ then

$$\sin(\gamma^c) = \frac{\dot{h}^d}{V_a} + \frac{1}{2\mathrm{m}gV_a}\Big(k_1\tilde{E}_K + k_2\tilde{E}_P\Big),$$

where $k_1 \triangleq |k_T - k_D|$ and $k_2 \triangleq k_T + k_D$. If $\dot{V_a}^d > 0$ then γ^c will increase which is the exact opposite of the desired behavior. If $0 < k_D = k_T$ then

$$\sin(\gamma^c) = \frac{\dot{h}^d}{V_a} + \frac{k_2\tilde{h}}{V_a}.$$

In this case, γ^c only depends on the desired climb rate and the altitude error and cannot take into account the coupling between the altitude and airspeed. Finally, if $0 < k_T < k_D$, then

$$\sin(\gamma^c) = \frac{\dot{h}^d}{V_a} + \frac{1}{2\mathrm{m}gV_a}\Big(-k_1\tilde{E}_K + k_2\tilde{E}_P\Big).$$

This case gives us the desired behavior because γ^c will increase if $\dot{V}_a^{\,d} < 0$ or $\dot{h}^{\,c} > 0$.

It is interesting to note that while the nonlinear thrust controller is very similar to the original TECS thrust controller, the flight path angle controllers have significant differences. First, the nonlinear controller uses the desired climb rate while the original controller uses the energy distribution rate. Furthermore, the nonlinear controller uses a weighted energy difference instead of the true energy difference. While using a weighted difference adds support to the modifications made in (6.11), the nonlinear controller has restrictions on the weighting. First, it is impossible to force the nonlinear controller to use the true energy difference because it would require $k_D = 0$. In addition, the potential energy error, $k_2 \in (k_T, 2k_T)$, is always weighted heavier than the kinetic energy error, $k_1 \in (0, k_T)$.

Unlike the original TECS controller, the nonlinear controller requires a desired airspeed, acceleration, altitude and climb rate. There are two different ways that the commands can be generated. The first approach is by using a reference model for the thrust and altitude where the desired climb rate and acceleration are generated by

$$\dot{h}^{\,d} = k_h\big(h^c - h^d\big) \tag{6.24}$$

$$\dot{V}_a^{\,d} = k_v\big(V_a^c - V_a^d\big). \tag{6.25}$$

The second approach uses the current state instead of the desired state to generate the desired rates which are computed by

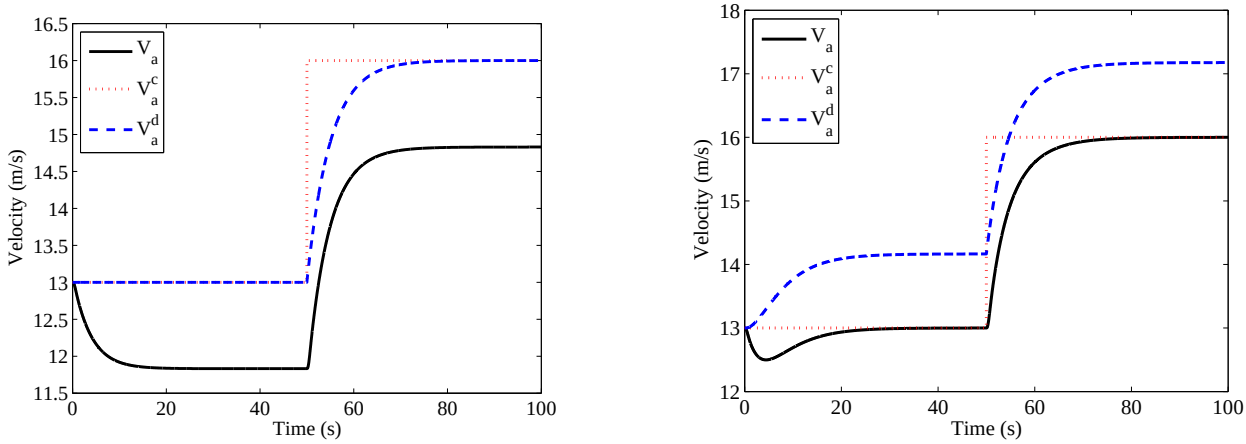
$$\dot{h}^{\,d} = k_h(h^c - h) \tag{6.26}$$

$$\dot{V}_a^{\,d} = k_v(V_a^c - V_a). \tag{6.27}$$

In both approaches the desired altitude and airspeed are computed by

$$h^d = h^d(t_0) + \int_{t_0}^{t} \dot{h}^{\,d}(\tau)\delta\tau$$

$$V_a^d = V_a^d(t_0) + \int_{t_0}^{t} \dot{V}_a^d(\tau)\delta\tau,$$



(a) No Feedback (6.25) **(b)** Feedback (6.27)

Figure 6.1: Comparison of the desired value generation methods

where the desired values are initialized as $h^d(t_0) = h(t_0)$ and $V_a^d(t_0) = V_a(t_0)$. Note that the desired acceleration and climb rate should have saturation limits.

One interesting feature of the second approach is that h^c does not need to equal h^d in steady state. However, we have noticed, in simulation, that the interaction between the changing desired values and parameter estimates can cause the nonlinear adaptive TECS controller to go unstable. This issue does not occur when the adaptive component is not used, and the second approach removes the steady state error for a step input. For example, Figure 6.1a shows the velocity response to a 3 m/s step command where the desired acceleration is generated by (6.25), where $k_v = 0.2$, and is saturated by ± 0.75 m/s^2. The aircraft is being controlled by the nonlinear TECS controller but the adaptive element is not being used. Note that the desired airspeed is a smoothed version of the step command and there is steady state error due to modeling errors. Figure 6.1b shows the same situation but the desired acceleration is being generated by (6.27). In this case, the desired airspeed increases until the error between the commanded and actual airspeed goes to 0. We will show that this is true in general for a linearized version of the system.

Consider the system shown in Figure 6.2 where the dynamics are [16]

$$\dot{h} = V_a \sin(\gamma)$$
$$\dot{V}_a = -g \sin(\gamma) - D + T,$$

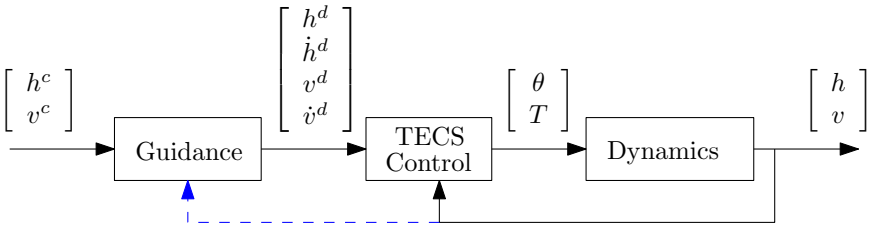


Figure 6.2: Structure for the nonlinear non-adaptive TECS controller. The guidance block, which generates the desired states, can have feedback represented by the dashed blue line.

the drag is given by

$$D = \frac{1}{2}\rho S V_a^2 C_{D_0}, \tag{6.28}$$

where $C_{D0} = 0.0439$, which assumes a constant angle of attack, the controller is (6.16) with $\boldsymbol{\phi} = 0$ and (6.20), and the guidance block can have feedback, given by (6.26) and (6.27), or no feedback, given by (6.24) and (6.25). Note that this ignores the pitch and thrust dynamics. These systems, both with and without feedback, were linearized about the operating point $\dot{h} = 0$ and $V_a = 13\ m/s$ with $k_T = 0.2$, $k_D = 0.25$, $k_h = 0.2$, and $k_v = 0.2$ providing the linear system

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}$$
$$\dot{\boldsymbol{y}} = \boldsymbol{C}\boldsymbol{x}.$$

Both systems are stable and the eigenvalues of the state space matrix $\boldsymbol{A}$ are

$$\text{eig}(\boldsymbol{A}_{\text{noFeedback}}) = [-0.2187 \pm j0.05,\ -0.2,\ -0.2]$$
$$\text{eig}(\boldsymbol{A}_{\text{feedback}}) = [-0.38,\ -0.25,\ -0.2 - 0.17].$$

If we define the error as

$$E(s) \triangleq R(s) - Y(s) = R(s) - \boldsymbol{C}(s\boldsymbol{I}_4 - \boldsymbol{A})^{-1}\boldsymbol{B}R(s),$$

where $R(s) = 1/s$ for a step input, then $H(s) \triangleq sE(s)$ is

$$H_{noFeedback}(s) = \begin{bmatrix} \frac{s\left(s^2+0.44s+0.05\right)}{d_{nf}(s)} & \frac{-0.004s+0.001}{d_{nf}(s)} \\ \frac{-s(0.08s+0.02)}{d_{nf}(s)} & \frac{s^3+0.33s^2+0.06s+0.008}{d_{nf}(s)} \end{bmatrix}$$

$$H_{feedback}(s) = \begin{bmatrix} \frac{s\left(s^3+0.75s^2+0.13s+0.002\right)}{d_f(s)} & \frac{s(-0.004s+0.001)}{d_f(s)} \\ \frac{s^2(-0.003s+0.001)}{d_f(s)} & \frac{s\left(s^3+0.64s^2+0.14s-0.01\right)}{d_f(s)} \end{bmatrix},$$

where

$$d_{nf}(s) = s^3 + 0.64s^2 + 0.14s + 0.002$$
$$d_f(s) = s^4 + 0.95s^3 + 0.28s^2 + 0.03s + 0.0003.$$

Note that the rows, of these matrices, correspond to the outputs h and V_a and the columns correspond to the inputs h^c and V_a^c. Applying the final value theorem, which states [39]

$$\lim_{t\to\infty} e(t) = \lim_{s\to 0} sE(s)$$

if the system is stable, by taking the limit of $H(s)$ as $s \to 0$ shows that the no feedback case has a steady state error in both altitude and airspeed due to the airspeed command and that adding feedback removes the steady state error completely.

6.4 Pitch and Thrust Control

Pitch and thrust control schemes are needed in order to implement the nonlinear TECS controller. In this chapter the pitch is controlled by a PID loop

$$\delta_e = k_{p,\theta}(\theta^c - \theta) - k_{d,\theta}q + k_{i,\theta}\int_{t_0}^{t}(\theta^c - \theta)\delta_\tau. \tag{6.29}$$

The thrust is modeled by [16]

$$T = \frac{1}{2}\rho S_{\text{prop}} C_{\text{prop}}\left(k_{\text{motor}}^2\delta_t^2 - V_a^2\right),$$

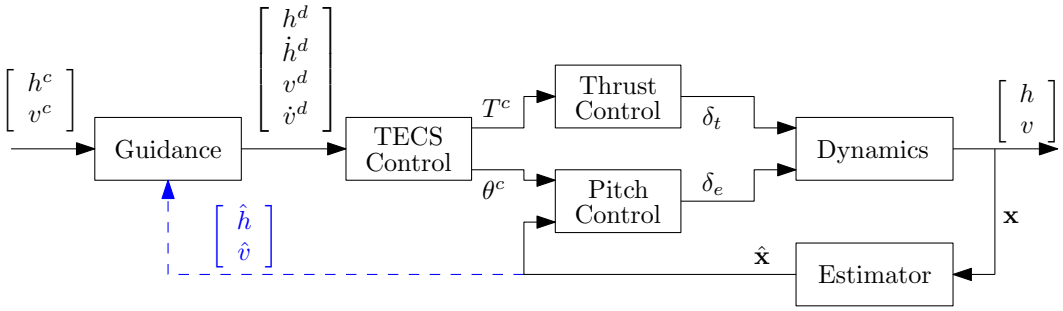


Figure 6.3: Realistic nonlinear non-adaptive TECS control architecture

where the parameters will be discussed in Section 6.7. This model can be simplified by combining terms to get

$$T = k_{T,1}\delta_t^2 - k_{T,2}V_a^2$$

which means the commanded thrust can be converted to a throttle command by

$$\delta_t = \sqrt{\frac{T^c + \hat{k}_{T,2}V_a^2}{\hat{k}_{T,1}}}, \tag{6.30}$$

where $\hat{k}_{T,1}$ and $\hat{k}_{T,2}$ are the estimates of the thrust parameters. Methods to estimate these parameters will be described later. The overall control architecture is shown in Figure 6.3 where the dashed blue line indicates the optional feedback in the guidance system.

One downside to this thrust controller is the lack of feedback. This is due to the lack of an accurate method to measure the thrust on small UAVs. Several different methods were explored to dynamically adapt these parameters based on the estimated drag and thrust and the measured acceleration. However, none of these methods worked well due either to the inaccuracies in the constant estimated drag parameters for the non-adaptive TECS controller or the dynamics of the estimated drag parameters for the adaptive TECS controller. However, it turns out that adding the feedback to the guidance block also corrects for inaccuracies in the thrust parameters. To show this we will perform a similar linearization analysis.

Consider the system

$$\begin{aligned}\dot{h} &= V_a \sin(\gamma)\\ \dot{V}_a &= -g\sin(\gamma) - D + T\\ T &= k_{T,1}\delta_t^2 - k_{T,2}V_a^2,\end{aligned}$$

where the drag and aerodynamic parameters are the same as the earlier example. Once again the estimated drag will be $\hat{D} = 0$. The flight path controller remains unchanged while the commanded thrust from (6.16) is converted to a throttle command using (6.30) where $\hat{k}_{T,1} = 1.3k_{T,1}$, $\hat{k}_{T,2} = 0$, $k_{T,1} = 7.964$, and $k_{T,2} = 0.199$.

Like the earlier example, both of the systems, with and without the guidance feedback, are stable. Computing $E(s)$, applying a step input, and multiplying by s provides

$$\begin{aligned}H_{noFeedback}(s) &= \begin{bmatrix} \frac{s(s^2+0.95s+0.16)}{d_{nf}} & \frac{-0.001s+0.005}{d_{nf}} \\ \frac{-s(0.03s+0.006)}{d_{nf}} & \frac{s^3+0.91s^2+0.3s+0.03}{d_{nf}} \end{bmatrix}\\ H_{Feedback}(s) &= \begin{bmatrix} \frac{s(s^3+1.2s^2+0.22s+0.001)}{d_f} & \frac{s(0.001s-0.005)}{d_f} \\ \frac{-s^2(0.03s+0.007)}{d_f} & \frac{s(s^3+1.16s^2+0.36s+0.03)}{d_f} \end{bmatrix},\end{aligned}$$

where

$$\begin{aligned}d_{nf} &= s^3 + 1.16s^2 + 0.236s + 0.03\\ d_f &= s^4 + 1.4s^3 + 0.46s^2 + 0.05s + 0.0002.\end{aligned}$$

Using the final value theorem we can conclude that adding feedback to the guidance block removes the steady state error even when no drag estimate is used and the thrust parameters are significantly off.

The nonlinear TECS controllers need to be able to estimate the angle of attack so that γ^c can be converted to a pitch command by $\theta^c = \gamma^c + \alpha$. If the angle of attack is not measured or estimated then the pitch command can be approximated by $\theta^c \approx \gamma^c + \alpha^t$, where α^t is the trim angle of attack at the nominal airspeed. A single α^t could be used or a

database of trim angles of attack for different airspeeds could be computed and used. This approximation reduces the overall accuracy of the nonlinear TECS control schemes.

6.5 Other Longitudinal Controllers

Three other altitude and airspeed controllers were implemented in order to provide a better comparison. The first controller consists of two PID loops on the total energy and the energy difference instead of the total energy rate and energy distribution rate that the original TECS controller uses. These control loops are given by

$$T^c = T_D + k_{p,T}\tilde{E}_T - k_{d,T}\dot{\tilde{E}}_T + k_{i,T}\int_{t_0}^{t} \tilde{E}_T \delta\tau$$
$$\theta^c = k_{p,\theta}\tilde{E}_D - k_{d,\theta}\dot{\tilde{E}}_D + k_{i,\theta}\int_{t_0}^{t} \tilde{E}_D \delta\tau.$$

The second controller is a standard decoupled successive loop closure control scheme where the thrust is controlled by a PI loop based on the airspeed error and the pitch is controlled by a PI loop based on the altitude error [16]. The commanded thrust and pitch are

$$T^c = T_D + k_{p,T}\tilde{V}_a + k_{i,T}\int_{t_0}^{t} \tilde{V}_a \delta\tau$$
$$\theta^c = k_{p,\theta}\tilde{h} + k_{i,\theta}\int_{t_0}^{t} \tilde{h} \delta\tau.$$

The final controller is a simplified version of the TECS controller used on the ArduPilot [92]. In this control scheme the thrust controller combines a feedforward term, given by

$$T_{ff} = T_D + k_{T,ff}\dot{E}_t^d + k_{T,\phi}\left(\frac{1}{\cos(\phi)^2} - 1\right),$$

where $k_{T,ff} > 0$ controls the feedforward amount and $k_{T,\phi}$ is a parameter that accounts for the increased drag while the plane banks, with a PID control loop based on the total energy error given by

$$T = T_{ff} + k_{p,T}\tilde{E}_t + k_{d,T}\dot{\tilde{E}}_t + k_{i,T}\int_{t_0}^{t} \tilde{E}_t \delta\tau.$$

The pitch controller uses a scaled version of the energy difference given by

$$E_3 \triangleq (2 - k_{ke})gh - \frac{k_{ke}}{2}V_a^2,$$

where $k_{ke} \in [0, 2]$ determines the weighting between the altitude and airspeed errors. A $k_{ke} = 0$ means that only the altitude error is used and $k_{ke} = 2$ means that only the airspeed error is used. Note that this scaled energy error is similar to the scaled energy error used in the nonlinear TECS controller and the most recent version of Lambregts' TECS controller [68]. The commanded pitch is generated by a PID loop with a feedforward term and is given by

$$\theta^c = \frac{1}{V_a}\left(k_{p,\theta}\tilde{E}_3 + \frac{\dot{E}_3^d}{g} + k_{d,\theta}\dot{\tilde{E}}_e + k_{i,\theta}\int_{t_0}^{t}\tilde{E}_3\delta\tau\right).$$

6.6 Lateral Control

A simple lateral control scheme was implemented based on the approach in [16] and a brief summary is presented here. The objective of the lateral controller is to track a commanded course, χ^c, which is the actual direction the aircraft is moving after taking into account the wind. Given the course error, a roll command is generated by

$$\phi^c = k_{p,\chi}\tilde{\chi} - k_{d,\chi}r + k_{i,\chi}\int_{t_0}^{t}\tilde{\chi}\delta\tau \tag{6.31}$$

which will attempt to drive the course error to zero. The roll error is used to generate the aileron command by

$$\delta_a = k_{p,\phi}\tilde{\phi} - k_{d,\phi}p + k_{i,\phi}\int_{t_0}^{t}\tilde{\phi}\delta\tau. \tag{6.32}$$

Note that (6.31) is saturated by $\pm\bar{\phi}$ to limit the commanded roll and (6.32) is saturated by ± 1.

The course command is generated by a vector field approach that drives the airplane to the straight line between two waypoints. Dubin's paths are used to smooth the transition between waypoints. See [9,16] for details about the path following and waypoint management methods.

6.7 Dynamics

These control schemes were tested on a Zagi airframe simulator from [16] instead of the V-Bat simulator. The kinematic equations are given by (3.1)-(3.4) where the net force is given by

$$\boldsymbol{f} = \boldsymbol{f}_g + \boldsymbol{f}_a + \boldsymbol{f}_t,$$

and the net moment is

$$\boldsymbol{m} = \frac{1}{2}\rho S V_a^2 \begin{bmatrix} b\Big(C_{l_\beta}\beta + \frac{b}{2V_a}\big(C_{l_p}p + C_{l_r}r\big) + C_{l_{\delta_a}}\delta_a + C_{l_{\delta_r}}\delta_r\Big) \\ c\Big(C_{m_0} + C_{m_\alpha}\alpha + C_{m_q}\frac{c}{2V_a}q + C_{m_{\delta_e}}\delta_e\Big) \\ b\Big(C_{n_\beta}\beta + \frac{b}{2V_a}\big(C_{n_p}p + C_{n_r}r\big) + C_{n_{\delta_a}}\delta_a + C_{n_{\delta_r}}\delta_r\Big) \end{bmatrix}.$$

The force of gravity is given by (3.5), the aerodynamic force is

$$\boldsymbol{f}_a = \frac{1}{2}\rho S V_a^2 \begin{bmatrix} -C_D\cos(\alpha) + C_L\sin(\alpha) + C_{L_{\delta_e}}\sin(\alpha)\delta_e \\ C_{Y_\beta}\beta + \frac{b}{2V_a}C_{Y_p}p + C_{Y_{\delta_r}}\delta_r \\ -C_D\sin(\alpha) - C_L\cos(\alpha) - C_{L_{\delta_e}}\cos(\alpha)\delta_e \end{bmatrix},$$

and the thrust from the motor is

$$\boldsymbol{f}_t = \begin{bmatrix} \frac{1}{2}\rho S_{\text{prop}} C_{\text{prop}}(k_{\text{motor}}^2\delta_t^2 - V_a^2) \\ 0 \\ 0 \end{bmatrix}.$$

The lift coefficient is

$$C_L = (1-\sigma(\alpha))(C_{L_0} + C_{L_\alpha}\alpha) + \sigma(\alpha)\big(2\,\text{sign}(\alpha)\sin^2(\alpha)\cos(\alpha)\big),$$

where

$$\sigma(\alpha) = \frac{1 + \exp^{-M(\alpha-\alpha_0)}\exp^{M(\alpha+\alpha_0)}}{(\exp^{-M(\alpha-\alpha_0)})(\exp^{M(\alpha+\alpha_0)})}$$

Table 6.1: Airframe parameters [16]

Parameter	Value	Parameter	Value	Parameter	Value
C_{L_0}	0.028	$C_{Y_{\delta_r}}$	-0.17	m	1.56 kg
C_{L_α}	3.45	C_{l_β}	-0.12	S	0.2589 m^2
$C_{L_{\delta_e}}$	0.36	C_{l_p}	-0.26	b	1.4224 m
M	50	C_{l_r}	0.14	c	0.3302 m
α_0	0.4712	$C_{l_{\delta_r}}$	0.08	C_{prop}	1 m
C_{D_0}	0.03	$C_{l_{\delta_a}}$	0.105	S_{prop}	0.0314 m^2
e	0.9	C_{n_β}	0.25	J_x	0.1147 kg m^2
C_{m_α}	-0.38	C_{n_p}	0.022	J_y	0.0576 kg m^2
C_{m_q}	-3.6	C_{n_r}	-0.35	J_z	0.1712 kg m^2
$C_{m_{\delta_e}}$	-0.5	$C_{n_{\delta_r}}$	-0.032	J_{xz}	0.0015 kg m^2
C_{Y_β}	-0.98	$C_{n_{\delta_a}}$	0.06	k_{motor}	20
C_{Y_p}	-0.26				

and M and α_0 are positive constants. The drag coefficient is

$$C_D = C_{D_0} + \frac{(C_{L_0} + C_{L_\alpha}\alpha)^2}{\pi e AR}, \tag{6.33}$$

where e is the Oswald efficiency factor and $AR = \frac{b^2}{S}$ is the aspect ratio of the wing. The airframe parameters, used in the simulation, are in Table 6.1.

6.8 Parameter Estimation

Drag and thrust estimates are needed in order to implement the nonlinear TECS controller with the open loop thrust controller (6.30). While these estimates could be generated using the model developed in Chapter 3, we will develop a simple parameter estimation scheme that only requires data from several flight tests and possibly a static bench test.

The lift, drag and thrust can be modeled by [16]

$$\begin{aligned} F_L &= \frac{1}{2}\rho S V_a^2 (C_{L_0} + C_{L_\alpha}\alpha) \\ F_D &= \frac{1}{2}\rho S V_a^2 (C_{D_0} + C_{D_\alpha}\alpha) \\ T &= \frac{1}{2}\rho S_{\text{prop}} C_{\text{prop}} \big(k_{\text{motor}}^2 \delta_t^2 - V_a^2\big). \end{aligned}$$

Note that the lift model is included because it is needed by the estimation algorithm. By combining terms these models can be simplified to

$$F_L = k_5 V_a^2 + k_6 V_a^2 \alpha \tag{6.34}$$

$$F_D = k_1 V_a^2 + k_2 V_a^2 \alpha \tag{6.35}$$

$$T = k_3 \delta_t^2 - k_4 V_a^2. \tag{6.36}$$

In steady level trim flight

$$F_L - \mathrm{m}g = T \sin(\theta) \tag{6.37}$$

$$F_D = T \cos(\theta). \tag{6.38}$$

Inserting (6.34)-(6.36) into (6.37) and (6.38) provides

$$k_5 V_a^2 + k_6 V_a^2 - \mathrm{m}g = \left(k_3 \delta_t^2 - k_4 V_a^2\right) \sin(\theta), \tag{6.39}$$

$$k_1 V_a^2 + k_2 V_a^2 \alpha = \left(k_3 \delta_t^2 - k_4 V_a^2\right) \cos(\theta). \tag{6.40}$$

By noting that $\dot{h} = 0$ implies $\gamma = 0$ and $\alpha = \theta$, the lift, drag, and thrust parameters can be estimated by collecting flight data for several different constant altitude trim conditions and solving the system of equations. If least squares is used then flight data from three different trim conditions are needed to provide estimates for all six parameters.

However, the TECS controller only needs the drag and thrust parameters. The parameter estimation problem can be simplified by only looking at (6.40). If θ is small then (6.40) can be approximated as

$$k_1 V_a^2 + k_2 V_a^2 \alpha = k_3 \delta_t^2 - k_4 V_a^2. \tag{6.41}$$

There is no way to determine the overall scale of the thrust and the drag from (6.41). The scale can be obtained by keeping the $\cos(\theta)$ term, however, the small changes in θ between the trim conditions are easily lost in noise. The scale ambiguity can be solved by performing some static bench tests in order to determine k_3. Another problem with (6.41) is that there

is no way to distinguish between the effects of $k_1 V_a^2$ of the drag and $k_4 V_a^2$ of the thrust. This can easily be corrected. Using these drag and thrust models, the throttle command is

$$\delta_t^2 = \frac{1}{k_3}\left(k_1 V_a^2 + k_2 V_a^2 \alpha + \boldsymbol{\phi}^\top \hat{\boldsymbol{\Psi}} + \frac{\dot{E}_T^d}{V_a} + k_T \frac{\tilde{E}_T}{V_a} + k_4 V_a^2\right). \tag{6.42}$$

The terms $k_1 V_a^2$ and $k_4 V_a^2$ can be combined into a single term $k_1' V_a^2$ where $k_1' \triangleq k_1 + k_4$. Using this, (6.41) becomes

$$k_1' V_a^2 + k_2 V_a^2 \alpha = k_3 \delta_t^2. \tag{6.43}$$

Adding the $\cos(\theta)$ back to (6.43), providing

$$k_1' V_a^2 + k_2 V_a^2 \alpha = k_3 \delta_t^2 \cos(\theta), \tag{6.44}$$

improves the accuracy slightly. Once again, k_1' and k_2 can be estimated by flying the plane at several different trim conditions, measuring the airspeed, pitch, and throttle command, and then performing least squares.

The parameter estimation methods were tested in simulation. Nine simulated flights were performed where the aircraft was in a steady level flight at nine different airspeeds ranging from 11 m/s to 15 m/s in 0.5 m/s increments, and the estimated airspeed and pitch and the throttle command were recorded. To reduce the effect of the noisy state estimates, the data were averaged over a length of time. The aerodynamic parameters k_1 to k_6 were then estimated, using these data, by an unconstrained optimization using (6.39) and (6.40) and optimization using (6.39) and (6.40) where k_3 was constrained to be within 30% of the true value. The control parameters, k_1', k_2, and k_3, were then computed from the aerodynamic parameters. In addition, the control parameters k_1', k_2, and k_3 were directly estimated using an unconstrained optimization using (6.44) where k_3 was set to the true value and an unconstrained optimization where k_3 using (6.44) was set to be 130% of the true value. The

Table 6.2: Mean, standard deviation, and maximum of the norm of the parameter error. A 30 second data averaging window and all nine airspeeds were used.

	Parameter Estimation Error		
Method	Mean (%)	STD (%)	Max (%)
Full - Unconstrained	172.6	1.55e-3	172.6
Full - Constrained Thrust	33.9	0.4	35.1
Simplified - True Thrust	34.0	0.4	35.3
Simplified - Thrust Error	45.4	0.3	46.3

control parameter estimates were then compared to the true values computed by

$$k_1' = \frac{1}{2}\rho S\left(C_{D_0} + \frac{C_{L_0}^2}{\pi e AR}\right) + \frac{1}{2}\rho S_{\text{prop}} C_{\text{prop}} \tag{6.45}$$

$$k_2 = \frac{1}{2}\rho S\left(\frac{2C_{L_0}C_{L_\alpha}}{\pi e AR} + \frac{C_{L_\alpha}^2 \alpha^t}{\pi e AR}\right) \tag{6.46}$$

$$k_3 = \frac{1}{2}\rho S_{\text{prop}} C_{\text{prop}} k_{\text{motor}}^2. \tag{6.47}$$

Note that first term in (6.45) and (6.46) come from the Taylor series expansion of (6.33) about the trim angle of attack α^t. All of the optimizations were performed using MATLAB's fmincon function. Each test was repeated 50 times with each test using a different random seed for the sensor noise.

During these tests, the accuracy and consistency of each method was tested along with testing different airspeeds flown and the length of the data averaging window. Table 6.2 compares the methods when the full range of airspeeds were used and the data were averaged over thirty seconds of flight. The full, unconstrained optimization found a minimum that matched the data but obtained unrealistic parameters that would not work in a real flight. However, constraining the thrust parameter allowed the full optimization to find parameters that were relatively close to the true values. Similar parameter estimates could be found using the simplified optimization if the true thrust parameter was known. However, the simplified optimization was very sensitive to inaccuracies in the thrust parameter. It is interesting that the optimization algorithms were so consistent in the parameter estimation error. This suggests that there are additional factors that could be taken into account to improve the accuracy.

Table 6.3: Mean, standard deviation, and maximum of the norm of the parameter estimation error. A 30 second data averaging window and the full constrained optimization were used.

Airspeeds (m/s)			Parameter Estimation Error		
Min	Increment	Max	Mean (%)	STD (%)	Max (%)
11	0.5	15	33.9	0.4	35.1
11	1	15	34.6	0.54	36.1
11.5	0.5	14.5	32.5	0.52	33.5
11.5	1	14.5	33.0	0.68	34.5
12	0.5	14	31.3	0.89	33.3
12	1	14	31.7	1.12	33.5

Table 6.4: Mean, standard deviation, and maximum of the norm of the parameter estimation error. All nine airspeeds and the full constrained optimization were used.

	Parameter Estimation Error		
Data Averaging Window (s)	Mean (%)	STD (%)	Max (%)
1	32.9	3.39	43.3
5	33.9	1.19	36.4
10	34.1	0.81	36.2
20	33.9	0.05	35.0
30	33.9	0.04	35.1

Table 6.3 explores how many trim conditions are needed when the full constrained optimization was used on data that were averaged over a thirty second window. The standard deviation of the parameter error increased as fewer airspeeds were included, however, the worst case and mean error decreased as the number of airspeeds included decreases. This is probably because the wider range of airspeeds had trim angle of attacks further from the value used to compute the true parameter. However, all of the airspeed ranges tested provided usable results.

Table 6.4 compares different data averaging window lengths when the full constrained optimization was used and the data included all nine airspeeds flown. As expected, averaging the data over a longer window reduced the standard deviation and the maximum error. In our opinion a data window between five and ten seconds provides the needed accuracy.

6.9 Simulation Results

The control algorithms were tested using an ideal and a realistic simulation. In the ideal simulation the pitch and thrust response are modeled by the second order dynamic systems

$$\ddot{\theta} = -2\zeta_\theta \omega_{n\theta} \dot{\theta} + \omega_{n\theta}(\theta^c - \theta)$$
$$\ddot{T} = -2\zeta_T \omega_{nT} \dot{T} + \omega_{nT}(T^c - T),$$

where $\zeta_\theta = \zeta_T = 0.707$ and $\omega_{n,\theta} = \omega_{n,T} = 5$. In addition, this simulation assumes that the controller has perfect state knowledge, that the dynamics are restricted to two dimensions, and that the estimated drag is $\hat{D} = 0.8D$ for both nonlinear TECS controllers. The realistic simulation uses an extended Kalman filter with simulated noisy sensors, see [16] for full details of the sensor models and the state estimation scheme, to generate the state estimates and the full three dimensional dynamics are used. In addition, (6.29) is used to control the pitch, the original and both nonlinear TECS controllers use the thrust controller given by (6.30), and a single α^t for $V_a = 13\ m/s$ is used because the estimation scheme does not estimate the angle of attack. The other control schemes' thrust controllers are modified to output a throttle command instead of a thrust command. The nonlinear TECS controllers use the parameter estimates,

$$\hat{k}_1' = 0.0248$$
$$\hat{k}_2 = 0.1131$$
$$\hat{k}_3 = 7.9614,$$

obtained by the full constrained optimization where $\hat{k}_3$ was constrained to be within 30% of the true value, the $12 : 0.5 : 14$ airspeeds were used, and the data was averaged for ten seconds. In every simulation the aircraft was initialized and commanded to fly at 13 m/s and $h = 100\ m$ for 50 seconds. This allowed all of the controllers to settle down and the parameter estimates, for the adaptive TECS, to enter steady state. Then altitude and/or

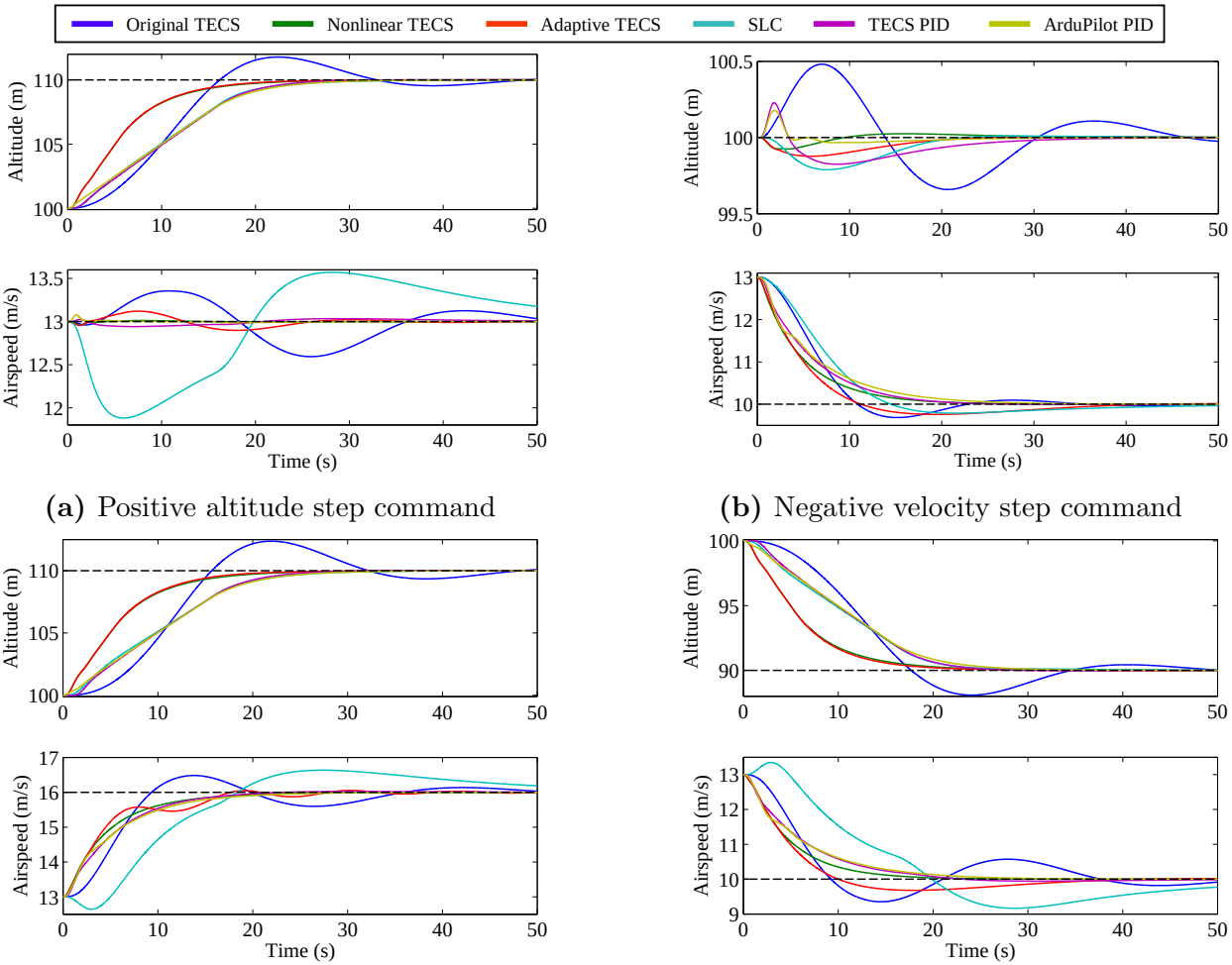
velocity step commands were given and the simulation ran for 50 more seconds. Only the last 50 seconds of flight are displayed.

Figure 6.4a shows the ideal response for a positive altitude step command while the desired velocity is constant and demonstrates the main weakness of the traditional decoupled successive loop closure control approach. Notice that when the desired altitude increases, the airspeed for the decoupled controller immediately starts to decrease. This issue is also shown in Figures 6.4c and 6.4d.

The original TECS controller does not have this problem because it accounts for the coupling in the dynamics. However, it tends to have the largest overshoot and, as shown in Figures 6.4a and 6.4b, the state that is commanded to be held constant oscillates while the other state undergoes the step command. The oscillation and overshoot can be reduced by using the TECS PID based on the energy instead of the PI controllers based on the energy rate. The ArduPilot TECS also handles these cases very well with only a slight altitude gain while decreasing the airspeed.

Both nonlinear TECS controllers are able to follow the commands very well and are able to achieve the fastest rise time with little to no overshoot. The adaptive TECS controller is able to correct for its incorrect drag model but it has a minor oscillation in its airspeed response. However, this oscillation is significantly less than the original TECS controller's oscillation. The oscillation can be completely removed by using the nonlinear controller without the adaptive element but with the guidance feedback.

The results of the realistic simulations for the exact same scenarios are shown in Figure 6.5. There are several interesting differences between the ideal and the realistic simulations. First, the original TECS controller's altitude response always has several meters of error in steady state which appears at the beginning and end of the simulations. This error is caused by the inaccurate thrust control scheme being used. Second, the adaptive TECS has steady state error in the airspeed and altitude when the aircraft is not traveling at 13 m/s. This steady state error could be removed, or reduced, by measuring the angle of attack or by using several different trim angles of attack. It is interesting that the non-adaptive TECS controller is able to overcome incorrect trim angle of attack and still achieve very nice performance in each case. The ArduPilot TECS control scheme is still able to achieve decent



(a) Positive altitude step command

(b) Negative velocity step command

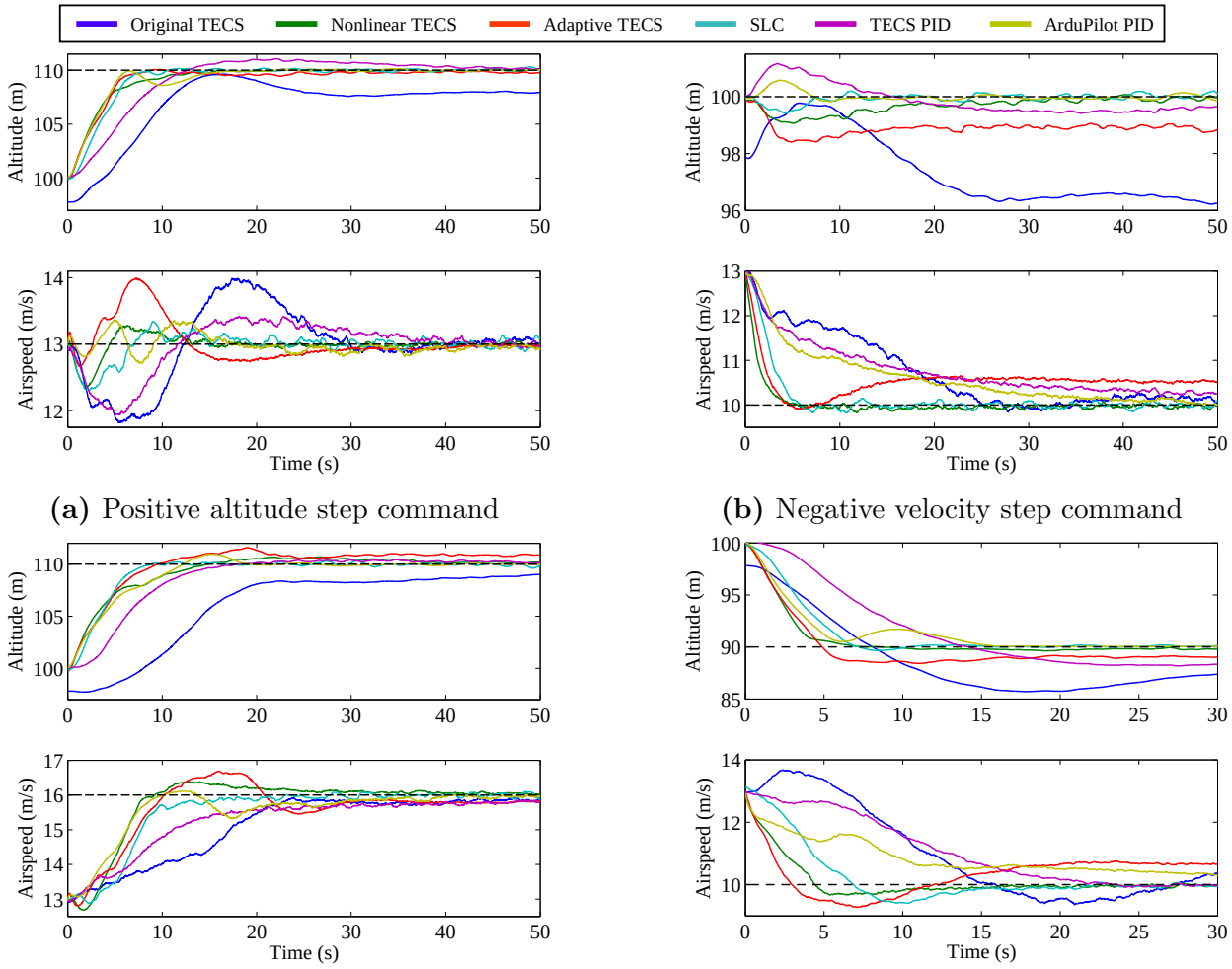
(c) Positive velocity and altitude step commands

(d) Negative velocity and altitude step commands

Figure 6.4: Ideal simulation results

performance, however, its rise time had to be significantly reduced due to its sensitivity to the noisy measurements. Finally, the TECS PID controller has a similar rise time as the original TECS control scheme yet it has a significantly smaller error.

Figure 6.6 shows two realistic simulations where the aircraft changes its heading by 180°. As expected the aircraft loses altitude during the turn with every control method. It is interesting that the ArduPilot has the largest variation in altitude and airspeed in Figure 6.6a even with the roll feedforward term. The successive loop closure loses a significant amount of airspeed in Figure 6.6b where the aircraft is climbing as well as turning. The nonlinear TECS controller has the best response in both tests.



(a) Positive altitude step command

(b) Negative velocity step command

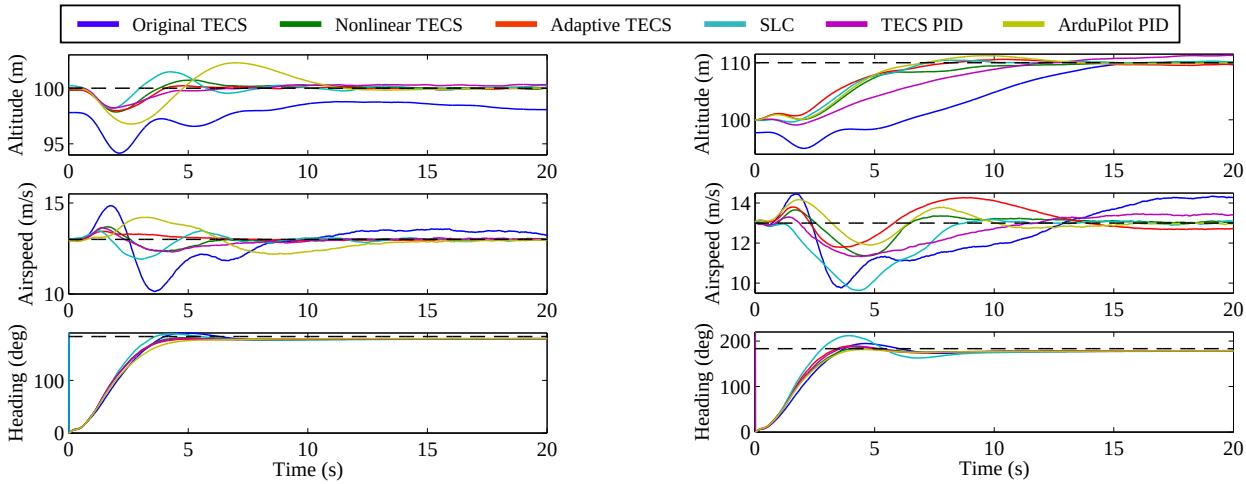
(c) Positive velocity and altitude step commands

(d) Negative velocity and altitude step commands

Figure 6.5: Realistic simulation results for wings level flight

6.10 Conclusions

In this chapter we derived a nonlinear altitude and airspeed controller for level flight based on TECS principles. The adaptive variant of this controller was shown to be able to track a desired airspeed and altitude using Lyapunov stability arguments and the non-adaptive variant with guidance feedback was shown to have zero steady state error for step inputs based on linearizing the system about a trim condition and applying the final value theorem. These nonlinear TECS controllers were compared with the original TECS control scheme as well as the standard successive loop closure, TECS PID, and the TECS controller used on the ArduPilot.



(a) Constant altitude and airspeed while turning

(b) Constant velocity while turning and climbing

Figure 6.6: Realistic simulation results for a banked flight

Simulation results, for the ideal system with perfect state knowledge and perfect thrust and pitch controllers, show that both nonlinear TECS variants handle the coupling between the altitude and airspeed dynamics and have a better response than the original TECS control scheme as well as the other control schemes. The realistic simulations, with noisy state estimates and realistic pitch and thrust controllers, show that even with inexact aerodynamic parameter estimates and an open loop thrust control scheme the nonlinear TECS controller has better performance than traditional control approaches. It should be pointed out that the realistic simulations assumed low cost noisy sensors and no aerodynamic model of the aircraft. Even with these assumptions the nonlinear TECS controller performed admirably. This control scheme's performance will only improve with better sensors, state estimation especially the angle of attack, a more exact drag model, and a thrust controller with feedback.

In addition, we presented a simple scheme to estimate the needed aerodynamic parameters using only some flight data and a simple bench test. These parameter estimates were used in the simulations which showed the method is accurate enough to generate useable parameters.

There are several aspects of this research that can be continued. First, this control scheme has only been tested for small, cheap UAVs. It would be interesting to see how

it performs on larger aircraft. Second, more advanced estimation schemes might be able to perform on-line estimation of the thrust and drag parameters based on the measured acceleration. This would improve the performance of the adaptive TECS controller. Finally, we would like to obtain stronger theoretical guarantees on the nonlinear TECS controller. The feedback in the guidance system is similar to what is done in the observer-like model reference adaptive control (OL-MRAC) scheme [69]. It would be interesting to see if any of the techniques developed for the OL-MRAC could be applied to the nonlinear TECS control problem. This might provide additional insight leading to stronger stability and tracking guarantees.