Online Parameter Estimation for the Adaptive Control of Unmanned Aerial Vehicles

Tristan Flanzer and S. Andrew Ning

December 10, 2009

1 Introduction

The accurate modeling of aircraft dynamics is essential when applying optimal control algorithms to unmanned aircraft. However, for low cost vehicles the dynamics may be difficult to predict. A number of factors work against the engineer; the aircraft is more likely to suffer manufacturing imperfections and rely on relatively crude actuation mechanisms, atmospheric disturbances play a significant role in performance, and under these conditions some aerodynamic performance parameters are difficult to accurately predict. Furthermore, a hard landing or other disturbance could easily alter the control surface trims of the aircraft and perhaps even it's dynamic response. Finally, low cost aircraft are more susceptible to actuator failures in flight. All of these factors motivate online parameter estimation. Our strategy is outlined as follows:

Repeat the following until objective is obtained (e.g. waypoint is reached) {

1. Use a Kalman filter and sensor data to provide estimate of aircraft state.

2. Perform actions based on current control gain matrix and current error in reaching desired state.

3. Every two seconds re-linearize the dynamic equations of motion about the current state and estimate control gains to maximize a quadratic reward function.

4. Every ten seconds make a maximum a posteriori estimate of aerodynamic parameters based on past states and prior knowledge of the parameters.

}

2 Unscented Kalman Filter

In an attempt to reproduce conditions in hardware, we use a special type of Kalman filter known as an Unscented Kalman Filter [1] to provide an estimate of the aircraft state based on noisy measurements. Like extended Kalman filters, UKFs allow estimation of non-linear functions. Kalman filters consist of two steps: prediction, followed by update. The prediction phase takes the previous state estimate and produces one for the current time step. In the update step, the current prediction is combined with current observation to refine the estimate. The state of the filter is represented by the a posteriori state estimate and a posteriori error covariance matrix. The UKF preserves much of this high level architecture. It differs in that the predict and update functions can be nonlinear, and that rather than linearizing the underlying model as done using an EKF, the UKF propagates a set of points through the nonlinear state and measurement functions and recovers an estimate of the mean and covariance of the state. The underlying intuition is that 'it is easier to approximate a probability distribution than it is to approximate an arbitrary nonlinear function or transformation'. Figure 1 shows output from a simulation demonstrating the accuracy of the state estimates. Here it is assumed that GPS position and velocity information is received at 4 Hz and that IMU data consisting of three axis accelerometer and gyroscopic data is used to advance the state estimate at 20 Hz.

3 Parameter Estimation

The objective of the parameter estimation is to learn a linear aerodynamic model for the aircraft. The motivation for this approach is that the aerodynamic forces of the aircraft (with the exception of drag) are well approximated by linear functions. The only nonlinearities that arise are in extremely rapid maneuvers or near stall. Our aircraft is not designed for aerobatic maneuvering and is not designed to fly

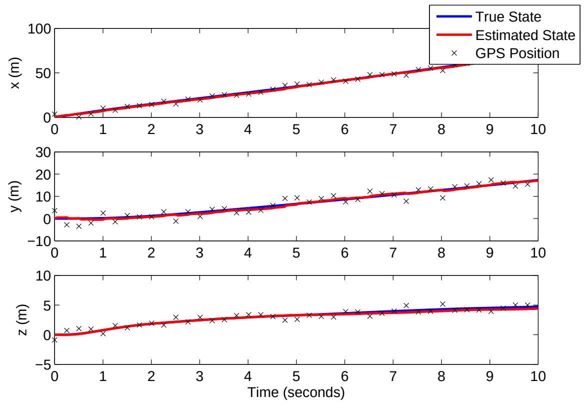

Figure 1: Unscented Kalman Filter estimate of aircraft horizontal, lateral, and vertical position.

close to its stall speed, allowing us to confidently use a linear model throughout the flight regime. Thus, learning an aerodynamic model is a more promising approach than trying to directly learn the dynamics of the aircraft which are inherently nonlinear.

The state vector for the aircraft is given by

$$s = (u, v, w, p, q, r, x, y, z, \phi, \theta, \psi)^T$$

where u,v,w are the velocities in the body frame, p,q,r are the angular velocities in the body frame, x,y,z are the positions of the aircraft in inertial space, and ϕ, θ, ψ are the Euler angles describing the orientation of the aircraft. The action vector for the aircraft is given by

$$a = (\delta_e, \delta_r, \delta_a, t)^T$$

where δ_e is the elevator deflection, δ_r is the rudder deflection, δ_a is the aileron deflection, and t is the throttle setting. The parameters of the aerodynamic model contain constant terms and the so called "stability derivatives" of the aircraft (the stability derivatives do not contain every possible term of a generic linear model since it is known that many of these terms are negligible for an aircraft). These parameters are arranged in to a column vector

$$\theta = (C_{L0} \dots \frac{dC_L}{d\alpha} \dots \frac{dC_n}{dp} \dots)^T, \qquad \theta \in \mathbb{R}^{25}$$

Only six of the state derivatives depend on these parameters, these we denote as

$$s_\theta = (u, v, w, p, q, r)^T$$

the other six state derivatives are functions only of the states and do not depend on the actions. We can rearrange the equations of motion as a stochastic linear function of the parameters θ

$$\dot{s}_\theta = A(s, a)\theta + b(s, a) + \epsilon \tag{1}$$

where A and b are nonlinear functions of the states and actions, and ϵ is assumed to be sampled from a Gaussian distribution ($\epsilon \sim \mathcal{N}(0, \sigma^2 I)$).

From our sensors we will not be able to provide $\dot{p}$, $\dot{q}$, and $\dot{r}$ directly. Numerical differentiation is also undesirable, since the state estimates will have some noise. Instead, we can use integration to relate $\dot{s}$ to the state in the next time step. Using a forward Euler approximation we have

$$s_{\theta t+1} = R(s_t, s_{t+1})[s_\theta + \dot{s}_\theta \Delta t]_t \tag{2}$$

where R is a rotation matrix that rotates from the body frame at time t to the body frame at time $t+1$ If we insert equation (1) into equation (2) and rearrange we have

$$s_{\theta t+1} = K_t\theta + c_t + \epsilon \tag{3}$$

where

$$K_t = R(s_t, s_{t+1})A(s_t, a_t)\Delta t$$

and

$$c_t = R(s_t, s_{t+1})(s_{\theta t} + b(s_t, a_t)\Delta t)$$

We will like to update our maximum likelihood estimation of θ periodically. Using all the state and action inputs from the previous update interval T-1 to the current time T as a training set, we can make a new estimate for θ_T. Before performing maximum likelihood we would like to incorporate some prior knowledge about the parameters. There are two motivations for doing this. First, from simulation we can often provide a reasonable starting estimate for the parameters θ. Second, we would like to avoid our training set growing larger and larger as time passes since we need to provide updates to our control strategy at a consistent rate. Thus, we will assume a prior distribution on θ of the form

$$\theta_T \sim \mathcal{N}(\theta_{T-1}, \tau^2 I)$$

where for $T = 1$, θ_0 is our initial estimate provided by the user. Thus, each update uses the previous update of θ as its prior.

Then the maximum a posterior estimate for θ is given by

$$\begin{aligned}
\theta_T &= \arg\max_\theta \prod_{t=T-1}^{T} p(s_{\theta t+1}|s_t, a_t, \theta)p(\theta) \\
&= \arg\max_\theta \sum_{t=T-1}^{T} \log p(s_{\theta t+1}|s_t, a_t, \theta) + \log p(\theta) \\
&= \arg\max_\theta \sum_{t=T-1}^{T} -\frac{1}{2\sigma^2}(K_t\theta - d_t)^T I(K_t\theta - d_t) \dots \\
&\qquad -\frac{1}{2\tau^2}(\theta - \theta_{T-1})^T I(\theta - \theta_{T-1})
\end{aligned}$$

$$= \arg\min_{\theta} \sum_{t=T-1}^{T} ||K_t\theta - d_t||^2 + \gamma||\theta - \theta_{T-1}||^2$$

where

$$d_t = s_{\theta t+1} - c_t$$

and

$$\gamma = \left(\frac{\sigma}{\tau}\right)^2$$

The optimization problem can be rearranged to the equivalent least squares problem

$$\theta_T = \arg\min_{\theta} \left\| \begin{bmatrix} K_{T-1} \\ \vdots \\ K_T \\ \sqrt{\gamma} I \end{bmatrix} \theta - \begin{bmatrix} d_{T-1} \\ \vdots \\ d_T \\ \sqrt{\gamma}\theta_{T-1} \end{bmatrix} \right\|$$

4 Control Strategy

We use reinforcement learning to choose the optimal control policy for piloting the aircraft. We choose a quadratic reward function, and use a linear model of the aircraft dynamics. The full nonlinear dynamics are available, but using a linear model greatly simplifies and speeds up the solution process. The optimization problem is given as:

$$\begin{aligned} \min_{a} \quad & \sum_{t=t_c}^{\infty} (s_t - s_d)^T Q (s_t - s_d) + a_t^T R a_t \\ \text{s.t.} \quad & s_{t+1} = A s_t + b a_t \end{aligned}$$

where s_d is the desired state, and Q and R are chosen according to Bryson's rule [2] as:

$$Q_{ii} = \frac{1}{\text{max acceptable value of } [(s - s_d)_i^2]}$$

$$R_{ii} = \frac{1}{\text{max acceptable value of } [a_i^2]}$$

This problem formulation is the infinite-horizon discrete linear-quadratic regulator (LQR) which has the solution of

$$a_t = -K(s_t - s_d)$$

with K being a matrix of optimal control gains. The nonlinear equations are re-linearized about the current state every two seconds, and the K matrix is updated. This allows us to still capture some of the nonlinear dynamical behavior of the aircraft.

5 Simulation and Aircraft Dynamics

A full six degree of freedom simulation was written in MATLAB to predict the behavior of an aircraft in flight. The aircraft equations of motion are integrated using a fourth-order accurate Runge-Kutta scheme. Aerodynamic forces and moments are assumed to be linear functions of the aircraft stability derivatives and aircraft state. The aircraft is assumed to have an elevator, rudder, ailerons, and single electric motor. The aerodynamic parameters are based on Mark Drela's Supra F3J sailplane, seen in Figure 2.

Figure 2: Supra F3J Sailplane

Some basic aircraft dimensions are listed below in Table 1.

Table 1: Aircraft Characteristics

Wing span	3.4 m
Wing area	0.667 m^2
Mass	1.36 kg
Cruise Speed	8 m/s

We assume that the aircraft is equipped with a GPS with a 4Hz update rate and an IMU that is queried at 20Hz. The GPS provides positions and velocities, while the IMU provides acceleration and angular velocities. The sensor uncertainties are shown in Table 2 and were based on commercially available inertial measurement units and GPS modules appropriate for this size of vehicle.

Table 2: Sensor Uncertainties

IMU Sensor Uncertainties	
x & y acceleration	0.002 g
z acceleration	0.0005 g
Heading angular rate	0.2 deg/s
Pitch & roll angular rate	0.06 deg/s
GPS Sensor Uncertainties	
x & y position	2.0 m
z position	6.0 m
x & y velocity	0.1 m/s
z velocity	0.3 m/s

6 Results

To test the effectiveness of the method we randomly initialize the aerodynamic parameters (θ) from a normal distribution with a mean equal to its true value but with a standard deviation of 50% of the parameter. This is a fairly large error; in practice we would expect to be able to provide a better starting point using aerodynamic analysis tools. However, we add the large uncertainty here to show robustness. In addition, one of the critical parameters $dc_l/d\delta_a$ has its sign changed. This parameter is the change in rolling moment with change in aileron deflection. Changing the sign of this parameter will cause the airplane to want to turn the wrong way. Finally, the following results assume a zero wind speed.

Figure 3 shows the path of the aircraft on a waypoint navigation mission. It starts at waypoint 0 and its objective is to pass through the other waypoints in order while maintaining a certain altitude and forward speed. We can see that initially the aircraft turns the wrong direction because we changed the sign of one of the parameters. However, it quickly learns the correct sign and is able to complete the mission successfully.

The other objectives were to climb to a steady state altitude of 5 m (relative to the starting altitude) at a forward speed of 8 m/s. Figures 4 shows the time history of the altitude and forward speed. At the beginning we can see that the altitude and forward speed are far from their desired values. The reason for this is that because of the parameter with the flipped sign, the accumulated error in heading angle gets larger and larger. Consequently that term in the LQR objective function becomes dominant and there is less focus on trying to minimize error in altitude or forward speed. As a better aerodynamic model is learned, the controller adapts and brings the aircraft to its steady state values.

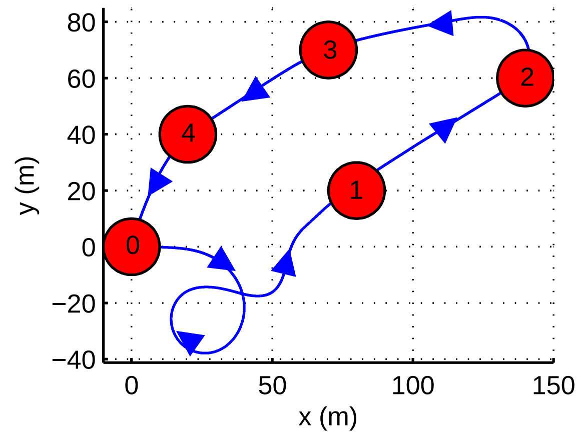

Figure 3: Top view of UAV path, waypoints denoted by circles

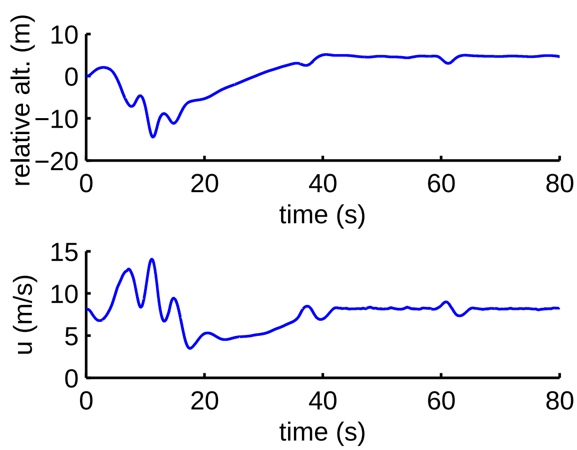

Figure 4: Time history of the relative altitude, and the forward speed (in body axes)

We need some metric to assess how well the supervised learning algorithm is doing in predicting the aerodynamic parameters. As mentioned, θ is updated every 10 seconds using the estimated states from that time interval and the prior estimate for θ. If θ were generating a good aerodynamic model to predict the next state, then we would expect that $||s_{\text{actual}} - s(\theta)||$ should be small for each time step in the time interval. Using our previous notation this is equivalent to the term $||K_t\theta - d_t||$ being small for $t = T - 1 \dots T$. Or, in other words we expect that the term

$$\alpha = \sqrt{\sum_{t=T-1}^{T} ||K_t\theta - d_t||^2}$$

should get smaller as the aerodynamic model is learned.

Figure 5 shows the change in this parameter α as a function of time. We see that in the first update, there is a large jump in performance. Most of this gain comes from correcting the parameter that started out with the wrong sign. In the subsequent 40 seconds the error is further diminished. For longer periods of time there is essentially no additional learning. There all several reason why the error will not go all the way to zero even in simulation. First, we are using estimates of the states rather than the true states and have included both sensor noise in the estimation. Second, we are using an Euler approximation for integrating from one time step to the next as discussed previously. Thus, α is a measure of the sum of the learning error, sensor noise, and numerical error.

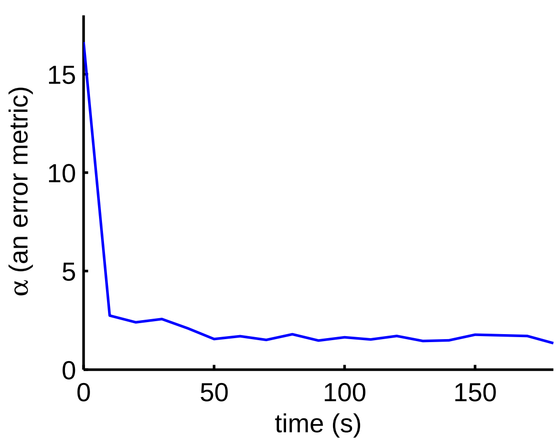

Figure 5: Convergence of the learned aerodynamic model to the true model as a function of time

7 Future Work

With the algorithm performing successfully in simulation, the next step will be to test its performance in hardware. A UAV is currently being designed for this purpose. A research autopilot [3] developed in the Aircraft Aerodynamics and Design Group will be used for autonomous control. The autopilot schematic can be seen in Figure 6. The sensor suite will include GPS, a 6 axis IMU, an airspeed sensor, and a barometric altitude sensor.

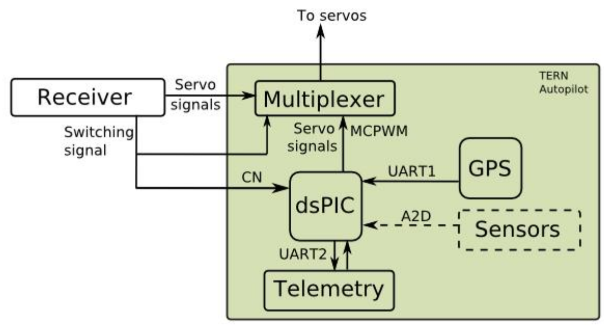

Figure 6: TERN Research Autopilot

References

[1] EA Wan and R. Van Der Merwe. The unscented Kalman filter for nonlinear estimation. In *The IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC*, pages 153–158, 2000.

[2] A.E. Bryson and Y.C. Ho. *Applied Optimal Control*. Wiley New York, 1975.

[3] C.K. Patel and I.M. Kroo. Theoretical and Experimental Investigation of Energy Extraction from Atmospheric Turbulence. In *26th International Congress of the Aeronautical Sciences*, 2008.