System Identification of Cessna 182 Model UAV

Ritchie Lee
Dept. of Aeronautics and Astronautics
Stanford University
Stanford, CA 94305
rcnlee@stanford.edu

Lu Shen
Dept. of Electrical Engineering
Stanford University
Stanford, CA 94305
lus@stanford.edu

Abstract
The first step to implementing an autopilot system on an Unmanned Aerial Vehicle (UAV) involves first to characterize the UAV's dynamics using a mathematical model. To accomplish this accurately for the particular UAV, the process of system identification, which is the estimation of the parameters of the equation of motion, is essential. However, experimental data is generally noisy and thus presents a challenging problem. In this research project, we are presented with several sets of flight test data of the UAV (a quarter scale model of a Cessna 182) and we wish to provide an accurate estimation of the system model parameters. Specifically, we apply and compare several different machine learning algorithms to the flight data including General Least Squares, Coordinate Descent, Recursive Estimation with Gaussian Noise, and Learn-Lagged-Linear. It is customary to evaluate a dynamic model by comparing its simulated state trajectories with experimental state trajectories. As a result, the dynamic models computed here are evaluated based on how well the simulated model trajectories compare with the flight data. Finally, an algorithm is suggested as a basis for in-flight updating of the system dynamics model in pseudo-real-time.

1 Vehicle Dynamical Model

In order to perform system identification, it is necessary to start first with an intended model of the system. Flight dynamics of fixed-wing aircrafts are reasonably well understood and there exist many literatures on the topic such as [1] and [2]. As suggested by [2], the dynamics in general are characterized by a set of non-linear differential equations in 8 states. However, under special circumstances and with specific assumptions, such as in the case of straight and steady flight, these equations of motion can be decoupled into lateral and longitudinal dynamics involving 4 states each. As given by [2], the lateral-direction equations of motion, linearized, decoupled under the assumption of small angle perturbation and zero cross product term $I_{xz}=0$, can be written in linear state-space form as

$$\frac{d}{dt}\begin{bmatrix}\Delta v\\ \Delta p\\ \Delta r\\ \Delta\phi\end{bmatrix}=\begin{bmatrix}Y_v & Y_p & -(u_0-Y_r) & g\cos\theta_0\\ L_v & L_p & L_r & 0\\ N_v & N_p & N_r & 0\\ 0 & 1 & 0 & 0\end{bmatrix}\begin{bmatrix}\Delta v\\ \Delta p\\ \Delta r\\ \Delta\phi\end{bmatrix}+\begin{bmatrix}0 & Y_{\delta r}\\ L_{\delta a} & L_{\delta r}\\ N_{\delta a} & N_{\delta r}\\ 0 & 0\end{bmatrix}\begin{bmatrix}\Delta\delta_a\\ \Delta\delta_r\end{bmatrix} \quad (1)$$

The longitudinal-direction equations of motion under the same assumptions are

$$\frac{d}{dt}\begin{bmatrix}\Delta u\\ \Delta w\\ \Delta q\\ \Delta\theta\end{bmatrix}=\begin{bmatrix}X_u & X_w & 0 & -g\\ Z_u & Z_x & u_0 & 0\\ M_u+M_{\dot{w}}Z_u & M_w+M_{\dot{w}}Z_w & M_q+M_{\dot{w}}u_0 & 0\\ 0 & 1 & 0 & 0\end{bmatrix}\begin{bmatrix}\Delta u\\ \Delta w\\ \Delta q\\ \Delta\theta\end{bmatrix}+\begin{bmatrix}X_{\delta e} & Y_{\delta T}\\ Z_{\delta e} & L_{\delta T}\\ M_{\delta e}+M_{\dot{w}}Z_{\delta e} & M_{\delta T}+M_{\dot{w}}Z_{\delta T}\\ 0 & 0\end{bmatrix}\begin{bmatrix}\Delta\delta_e\\ \Delta\delta_T\end{bmatrix} \quad (2)$$

where $[\phi\ \theta\ \psi]$ is the roll, pitch, yaw Euler angles, $[p\ q\ r]$ is the angular velocity of body-fixed axis frame relative to earth-fixed frame, $[u\ v\ w]$ is the linear velocity of the body-fixed axis frame, and $[\delta a.\ \delta r\ \delta e\ \delta t]$ are the aileron, rudder, elevator and throttle inputs, respectively. For compactness, each of (1) and (2) can be rewritten in the form

$$\dot{x} = Ax + Bu \quad (3)$$

2 General Least Squares

In this approach, we attempt to find the best estimate of the unknown parameters in A and B by applying least squares regression to the following objective function

$$||\dot{x} - (Ax + Bu)||_2^2 \quad (4)$$

To do this, the data is rearranged into the conventional form

$$\left|\left|\hat{y} - \hat{A}\hat{s}\right|\right|_2^2 \quad (5)$$

where the state data and unknown parameters are reordered as $\hat{A}$ and $\hat{s}$, respectively. In this form, the pseudo-inverse is used to find $\hat{s}$.

While the model results are fair, it leads to the notion that the proper objective function is not being considered since the model is evaluated based on its simulated trajectory.

3 Coordinate Descent

Since we are evaluating the performance of the dynamical model based on its simulated trajectory relative to the actual trajectory, a reasonable objective function would be to minimize the mean squared error between the simulated states and actual states at each time step. Numerically minimizing the cost function for each parameter one at a time and iterating through the parameters in random permutations, we hope to be able to converge to a reasonable objective function minimum.

However, initial trials suggest that this is a very computationally expensive algorithm that does not scale well with number of parameters and trajectory lengths, and therefore is prohibitively expensive for in-flight computation.

4 Recursive Estimation with Gaussian Noise

A recursive algorithm for the optimal estimation of a stationary state vector assuming zero mean Gaussian noise is presented in [3]. The two recursion equations are

$$Q_{k+1} = Q_k - Q_k A_{k+1}^T (A_{k+1} Q_k A_{k+1}^T + \Sigma_{k+1})^{-1} A_{k+1} Q_k \quad (6)$$

$$\hat{s}_{k+1} = \hat{s}_k + Q_k A_{k+1}^T (A_{k+1} Q_k A_{k+1}^T + \Sigma_{k+1})^{-1} (y_{k+1} - A_{k+1}\hat{s}_k) \quad (7)$$

where $\hat{s}_k$ and Q_k are the optimal state and covariance estimates given k measurements y_1 to y_k. Using a similar rearrangement as in (5), the unknown parameters are posed as the estimated state $\hat{s}_k$ and the data at time k is reordered as A_k. Recursive methods are ideal for in-flight computations since it does not require the reconsideration of past data. All past data information is embodied in the most recent state and covariance estimates.

However since the prior covariances of the data and parameters are unknown, we need to devise a method to estimate these. The initial estimate on the parameters was chosen to be all zero with relatively large decoupled covariances to allow for the presented data to bear more weight than the priors. Also, since the covariance of each individual data point is unknown, the measurement covariance of each state was assumed constant with cross-terms assumed to be zero. With these assumptions, the simulation results perform reasonably well, however, it leads us to the notion that we are not optimally estimating the proper covariances.

5 Learn-Lagged-Linear

A method to do well on this simulation criterion without the estimation of covariances is to learn the model parameters by optimizing the following "lagged criterion" as suggested by [4]:

$$\sum_{t=1}^{T-H}\sum_{h=1}^{H}\left|\left|\hat{s}_{t+h|t}-s_{t+h}\right|\right|_2^2 \quad (8)$$

However, this is in general a non-linear and difficult problem that leads to prohibitively expensive algorithms such as the EM algorithm suggested in [4]. Motivated by this, an algorithm to approximately minimize the lagged criterion is given by [5] called Learn-Lagged-Linear. For convenience, it is excerpted here.

LEARN-LAGGED-LINEAR:

1. Use least squares to minimize the one-step squared prediction error criterion to obtain an initial model $A^{(0)}, B^{(0)}$. Set $i = 1$.
2. For all $t = 1, \ldots, T$, $h = 1, \ldots, H$, simulate in the current model to compute $\hat{s}_{t+h|t}$.
3. Solve the following least squares problem:
   $$(\bar{A}, \bar{B}) = \arg\min_{A,B} \sum_{t=1}^{T-H}\sum_{h=1}^{H} \|(s_{t+h} - s_t) - (\sum_{\tau=0}^{h-1} A\hat{s}_{t+\tau|t} + Bu_{t+\tau})\|_2^2.$$
4. Set $A^{(i+1)} = (1-\alpha)A^{(i)} + \alpha\bar{A}$, $B^{(i+1)} = (1-\alpha)B^{(i)} + \alpha\bar{B}$.[3]
5. If $\|A^{(i+1)} - A^{(i)}\| + \|B^{(i+1)} - B^{(i)}\| \le \epsilon$ exit. Otherwise go back to step 2.

A horizon time H of 2 seconds (20 data points) was chosen. A comparison of the results is presented in Figure 1.

6 An Online Learning Algorithm Using LLL

Applying online learning concepts to the LLL algorithm leads to an algorithm suitable to serve as a basis for in-flight updating of the system dynamics model in pseudo-real-time.

The algorithm alternates between an evaluation step and a least squares step that solves for the new parameters.

ONLINE LEARNING USING LLL:

1. Start with the model found by running the LLL algorithm on available training data (mentioned in Section 5).
2. Simulate the current model on the given new test data.
3. Compare the simulated state trajectories with the actual measured state data at each specific time step and calculate the state error.
4. At each time step, if the calculated errors in the previous consecutive N time steps are all above a preset tolerance threshold, reiterate a new model using the LLL algorithm using the following input parameters:
    a. If the current time step is less than a lower bound L, then the algorithm uses all of the training data and the simulated data from all previous time steps for re-iteration.
    b. If the current time step is larger than the lower bound L, but less than an upper bound U, then the algorithm uses all available simulated data and part of the training data for re-iteration. The size of the training data used in the re-iteration can be adjusted based on, for example, size of the available simulated data.
    c. If the current time step is larger than the upper bound U, then the algorithm uses all available simulated data for re-iteration.
5. Continue the simulation and repeat steps 2 through 4 as necessary.

This algorithm will continually adapt to the new test data and prevent the model and new measurement data from having diverging state trajectories. The final results of the algorithm are compared here in Figure 1:

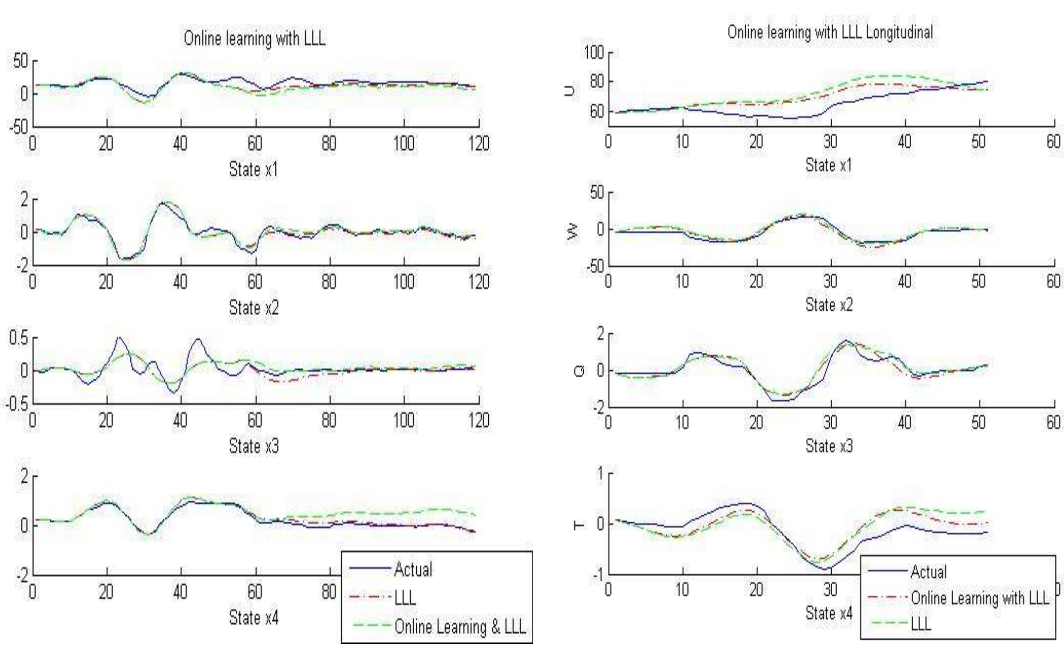

Figure 1: A Comparison of Lateral and Longitudinal Simulated Trajectories for Online Learning Using LLL versus LLL without Online Learning

7 Summary and Future Work

In this literature, we have discussed several different methods for accomplishing system identification of a UAV: Generalized Least Squares, Coordinate Descent, Recursive Estimation, and Learn-Lagged-Linear. Considering speed and performance, the Learn-Lagged-Linear algorithm presents itself as the best candidate for finding the model that performs well in the simulation criterion. A summary of the results are shown here in Table 1.

Table 1: A Comparison of State Errors with Algorithm

	State 1	State 2	State 3	State 4	State 5	State 6	State 7	State 8
LS – Train	20.45	1.07	0.82	0.55	72.77	23.11	3.76	0.91
LS - Test	76.11	1.67	1.10	1.77	112.88	48.56	11.81	3.30
Recursive - Train	20.12	1.04	0.81	0.52	72.56	23.09	1.36	0.92
Recursive - Test	72.31	1.81	1.08	2.33	109.26	45.61	10.71	3.90
LLL - Train	19.6	1.04	0.79	0.51	71.93	23.07	1.20	0.90
LLL - Test	71.06	1.75	1.04	3.15	107.99	45.05	4.17	3.88

	State 1	State 2	State 3	State 4	State 5	State 6	State 7	State 8
Online Learning with Least Square	64.86	2.02	1.07	1.41	110.16	57.15	5.74	2.98
Improvement (with and without learning)	14.78	-20.95	2.72	20.33	2.40	-17.68	1.39	59.69
Online Learning with LLL	54.18	1.71	1.07	0.86	93.67	43.11	4.10	3.56
Improvement (with and without learning)	23.75	2.28	2.88	72.69	13.26	4.30	1.67	8.24

Although the online learning algorithm presented in Section 6 performs reasonably well on the given dataset, testing on more extensive data is recommended to develop robust criteria that will perform well under a wide range of aircraft states. The algorithm is also fairly computationally expensive which renders it unsuitable for fully real-time updates. Future projects may involve the development of faster algorithms as well as an online learning algorithm that is robust enough to reliably detect the transitioning from accelerating flight, to straight and steady flight, to a coordinated banked turn.

Acknowledgements
We give warm thanks to Corey Ippolito, Pieter Abbeel, and Adam Coates for their guidance on this work.

References

[1] J. H. Blakelock. Automatic Control of Aircraft and Missiles. Second Edition. John Wiley & Sons, Inc. 1991.
[2] R. Nelson. Flight Stability and Automatic Control. McGraw Hill Book Company, New York, 1995
[3] S. Lall. ENGR207b Introduction to Modern Control. Lecture Notes. Stanford University 2006.
[4] P. Abbeel and A. Y. Ng. Learning First Order Markov Models for Control. In NIPS 18, 2005.
[5] P. Abbeel, V. Ganathi and A. Y. Ng. Learning Vehicular Dynamics, with application to modeling helicopters. In NIPS 2005.