A Machine Learned Model of a Hybrid Aircraft

Brandon Jones, Kevin Jenkins

CS229 Machine Learning, Fall 2016, Stanford University

I. Introduction

Aircraft development programs rely on aircraft dynamic models for flight control design, handling qualities analysis, and vehicle simulation [1]. Expensive flight tests are performed to validate the aircraft dynamic models with respect to the as-built aircraft, and to quantify unforeseen vehicle flight issues. Flight tests typically consist of targeted system identification flight maneuvers that are performed while vehicle state data is collected [2]. Industry methods rely heavily on domain knowledge of aircraft dynamics for manual piloted data generation and feature identification [3]. This project aims to simplify the modeling process with a Machine Learning pipeline that generates a non-linear vehicle model with minimal reliance on domain knowledge using full datasets throughout an entire flight envelope. A hybrid Vertical Takeoff and Landing (VTOL) aircraft with a fixed-wing with vertical and forward hover motors is used as a case study. This aircraft configuration is particularly suitable to a Machine Learning approach due to complex aerodynamic interactions and wide ranging airspeed envelope which does not fit standard classical aircraft models.

A. Background

The dynamics of an aircraft can be most generally expressed as the nonlinear first-order differential equation,

$$\dot{x} = f(x, u) \tag{1}$$

with vehicle state x, and the vehicle state derivative $\dot{x}$, and actuator input u. Understanding the function f for a given aircraft is critical for airframe design, control law design, and simulation. Therefore, a significant amount of effort is spent studying and modeling the function f for a specific aircraft.

B. Previous Work

The discipline of extracting the model $f(u, u)$ from data is referred to as "system identification" [4]. Specific system identification methods as applied to aircraft have been studied and applied for nearly 100 years [2]. However, current industry techniques still require domain knowledge, hand analysis, and manual feature selection to achieve acceptable performance and are generally airframe configuration specific. Methods employed include linear regression and maximum likelihood methods to define a linear model of an aircraft [5]. Well established locally linear model structures for conventional aircraft consist of rigid-body equations of motion and aerodynamic forces and moments [6] and can be summarized by the Linear Parameter Varying State-Space model,

$$\dot{x} = A(V_a, \rho, \alpha, \beta)x + B(V_a, \rho, \alpha, \beta)u. \tag{2}$$

The state-space matrices A and B are "scheduled" by airspeed V_a, air density ρ, angle-of-attack α, and sideslip β. These formulations provide tangible insight into the aerodynamics of an aircraft, however they may fail to capture non-linear dependencies on other parameters. A method to identify the Linear Parameter Varying state-space models was investigated using an Expectation-Maximization technique [7], however this method required specifically choosing scheduling parameters. A global, real-time, nonlinear aircraft modeling technique was presented that leveraged fuzzy logic and multivariate orthogonal functions to define relationships between a collection of identified locally linear models [8]. This approach identified a model across a flight envelope, but relied on a-priori formulation of aerodynamic coefficients and equations of motion. All these approaches are considered parametric regression methods in that they can be described by a finite number of parameters. A significant comparison between classical system identification paradigms of maximum likelihood and prediction error methods with the Machine Learning kernel-based regularization techniques was performed in [9]. This showed that learning techniques tailored to specific features of dynamic systems can outperform conventional parametric approaches for identification of linear systems.

II. Data

The reference training and test data was generated from an existing non-linear, high-fidelity aircraft flight simulation that included rigid body dynamics, aerodynamics, and actuator models. A total of 37 vehicle states were recorded and are described in Table I. These states were selected with little regard to the relevance to demonstrate an algorithm that does not require specific domain knowledge to provide adequate predictions. For example, motor torques, speeds, and thrusts were selected which are highly correlated states.

During normal cruise or hover flight the vehicle may be at a static flight condition resulting in large portions of collinear datapoints. Furthermore, due to the augmented nature of the flight controls system states can be highly correlated. Although it is an objective to be robust to these factors, a test maneuver capability was added to the vehicle flight control system software to stochastically command arbitrary actuator inputs to ensure the vehicle dynamics were observable. A total of 34 minutes of simulated flight data across the airspeed envelope from hover to cruise flight was collected.

III. Method

The problem of identifying an aircraft dynamics model can be described as a supervised learning regression problem. The data is formulated into a Markov Decision Process framework and a non-parametric method is described that is based on Locally Weighted Linear Regression[10]. An advantage of Locally Weighted Linear Regression is it generates a locally linear and parametric model at a particular state. This allows the option to extract a locally linear model for control stability analysis or manual inspection of the dependance on states. Only the longitudinal states of the aircraft are considered for this investigation to illustrate the regression algorithm.

A. Model

The generalized equation describing aircraft motion, 1 can be formulated as a discrete Markov Decision Process,

$$x^{(t+1)} = f(x,u) \cdot x^{(t)} + g(x,u) \cdot u^{(t)} \tag{3}$$

with state $x^{(t)} \in \mathbf{R}^{15\times m}$, actuator input $u^{(t)} \in \mathbf{R}^{22\times m}$. The objective is to learn the non-linear functions $f(x,u)$ and $g(x,u)$ to minimize the prediction error of $x^{(t+1)}$. The training examples $(x^{(i)}, y^{(i)})$ are stacked into the matrices,

$$X = \begin{bmatrix} x^{(t)} & \dots & x^{(m-1)} \\ u^{(t)} & \dots & u^{(m-1)} \end{bmatrix}$$
$$Y = \begin{bmatrix} \alpha^{(t+1)} & \dots & \alpha^{(m)} \\ \theta^{(t+1)} & \dots & \theta^{(m)} \\ \dot{\theta}^{(t+1)} & \dots & \dot{\theta}^{(m)} \\ V_{\text{air}}^{(t+1)} & \dots & V_{\text{air}}^{(m)} \end{bmatrix} \tag{4}$$

with input features X and target variables Y consisting of longitudinal states only, angle of attack α, pitch θ, pitch rate $\dot{\theta}$, and airspeed V_a.

B. Selectively Weighted Linear Regression

A Selectively Weighted Linear Regression algorithm is proposed. The typical locally-weighted linear regression is used to find the parameter $\theta \in \mathbf{R}^{n_y \times n}$ such that,

$$\theta = \underset{\theta}{\arg\min} \left(\sum_i w^{(i)} \left\| Y^{(i)} - \theta^T X^{(i)} \right\|^2 \right) \tag{5}$$

with weights,

$$w^{(i)} = \exp\left(-\frac{\left\| X^{(i)} - X \right\|^2}{2\tau^2} \right). \tag{6}$$

This technique uses the same vector X to establish weightings and to form the regression coefficients in θ. Selectively weighted regression, by contrast, uses feature selection to separate states that are important for local weighting from those with strong regression information. All elements of X that are not helpful for weighting or regression can re removed, then removes elements individually from X_r and X_w, to be used in the new minimization for $\theta \in \mathbf{R}^{n_y \times n_r}$,

$$\theta = \underset{\theta}{\arg\min} \left(\sum_i w^{(i)} \left\| Y^{(i)} - \theta^T X_r^{(i)} \right\|^2 + \lambda \left\| \theta \right\|_2^2 \right) \tag{7}$$

with weights,

$$w^{(i)} = \exp\left(-\frac{\left\| X_w^{(i)} - X_w \right\|^2}{2\tau^2} \right), \tag{8}$$

and an L_2 regularization constant λ. The L_2 regularization term penalizes large values of θ and improves machine precision issues with linearly-dependent columns. The training samples X_r and X_w contain selected elements from X,

$$\begin{aligned} X_w^{(i)} \in R^{n_w} &= \begin{bmatrix} X_{j_1}^{(i)} & X_{j_2}^{(i)} & \dots & X_{j_{n_w}}^{(i)} \end{bmatrix}^T \\ X_r^{(i)} \in \mathbb{R}^{n_r} &= \begin{bmatrix} X_{k_1}^{(i)} & X_{k_2}^{(i)} & \dots & X_{k_{n_r}}^{(i)} \end{bmatrix}^T \end{aligned} . \tag{9}$$

Removing specific features may allow for a more lightweight algorithm that can use fewer coefficients and operate more quickly. It also allows for more precise weighting by not down-weighting training samples that are far from the test entry on un-important dimensions. Two methods of feature selection are presented,

- Principle Component Analysis was run on the features X, and only the top linearly uncorrelated variables are selected for the regressors, X_r
- A hand "smart" selection of features corresponding to only the longitudinal axes of the vehicle was selected for X_w. For example, features on the roll

Airplane States (x15)	Airplane Actuators (x22)
Airspeed	Vertical and Forward Motor Thrust (x6)
Air Density	Vertical and Forward Motor Torque (x6)
Angle of Attack	Vertical and Forward Motor Speeds (x6)
Attitude (x3)	Aerodynamic Control Surface Deflection (x4)
Rotational Acceleration (x3)	
Body Acceleration (x3)	
Velocity (x3)	

Table I
AIRCRAFT STATES

and yaw axis were discarded. It would be the expectation that the hand selected feature set would predict the longitudinal states with less error due.

IV. RESULTS

A. Algorithm Performance

A summary of the training and test error results from four Locally Weighted Linear Regression methods are presented in Table II. A total of 34 minutes of data was used, where 70% of the data was used for training and 30% used for test. Specific τ and λ were chosen based on parameter sweeps across a range of values. Surprisingly, regression methods that did not involve removing features worked best. In particular, the "smart" removal of states resulted in the worst performance, highlighting potential unintuitive vehicle dynamics where longitudinal states have dependence on lateral directional states.

The results show the L_2 regularization did not improve the overall test error, and had worse training error for small sample sets. This result was particularly surprising given L_2 regularization should help the singularly issues with the many collinear states in the parameter sets during long straight and level flight segments.

B. Training Size

A comparison between each method versus training size is presented in Figure 1. In general the training errors converged to roughly a stable error threshold. However, the locally weighted methods showed potential for further decrease in error with a larger training set. The larger test error could be evidence of high variance, where additional training examples would reduce error.

C. Feature Removal

The "smart" removal of states resulted in less error with fewer training examples, however the error did not improve with more training examples. This demonstrates that this method did not leverage the non-standard dynamics of the hybrid aircraft observed in a more general training set. The large error and small gap between test and training error is evidence of high bias. Removal of features using PCA demonstrated lower error with fewer training examples, however was not able to outperform the methods that included all features.

D. Time Series Results and Error Level Verification

An example of the one-step performance of Locally Weighted Linear Regression on the dataset with $\tau = 50$ and $\lambda = 10^{-3}$ is presented in Figure 2. This result shows reasonable predictions of angle of attack, pitch, pitch rate, and airspeed, partially during periods with lower dynamics. During the period of higher maneuvering, from 1 sec to 4 sec, prediction errors increased as much as 10 deg/sec in pitch rate, however the other states were predicted well. Overall this performance would be reasonable for simulation analysis.

E. Numerical Solution Considerations

1) Normal Equation: To solve the optimization problem in Equation 7, the normal equation,

$$\theta = (X_r W X_r^T + \lambda)^{-1}(X_r W Y^T) \qquad (10)$$

was used. Unfortunately, since $W \in \mathbf{R}^{m\times m}$ as the training samples grow directly solving Equation 10 is $\approx \mathcal{O}(m^{2.3})$. As a workaround, training examples that resulted weightings $< 10^{-10}$ were removed which improved machine-precision linear independence of columns. However, the remaining columns were fairly similar resulting in matrices with low condition numbers.

2) Gradient Descent: To overcome the complexity increase, low condition numbers, and to allow Locally Weighted linear regression to scale to larger datasets, two gradient descent approaches were investigated. Generally, the gradient of the Locally Weighted linear regression loss function can be formulated as,

$$\nabla_\theta J(\theta) = 2\left(X_r^\intercal W_{i,i} X_r \theta - X_r^\intercal W_{i,i} Y\right). \qquad (11)$$

3) Stochastic Gradient Descent : Stochastic Gradient Descent was implemented using the update rule,

$$\theta \leftarrow \theta + \alpha(X_r^{(i)\intercal} W_{i,i} X_r^{(i)} \theta - X_r^{(i)\intercal} W_{i,i} Y^{(i)} \qquad (12)$$

where i is a random training example. This technique demonstrated to be very sensitive to choice of α, requiring tuning for robustness across different test samples

Linear Regression Method			**Training**		**Test**	
	λ	τ	**Error**	**Samples**	**Error**	**Samples**
Locally Weighted	0	50	0.55	15120 (25 minutes)	4.13	5537 (9.23 minutes)
Locally Weighted	10^{-3}	50	0.55		4.13	
Weighted with PCA	10^{-3}	50	1.3		4.38	
Weighted "smart removal"	10^{-3}	50	16.89		10.92	

Table II
COMPARISON OF LOCALLY WEIGHTED LINEAR REGRESSION METHODS

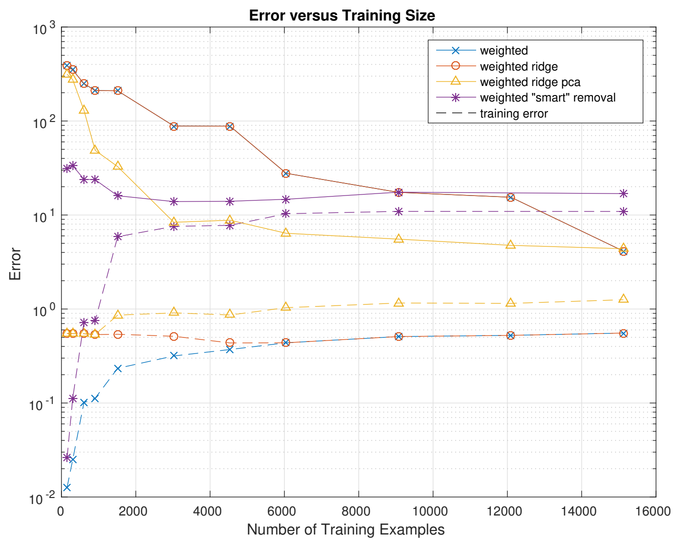

Figure 1. Training and Test Error Comparison

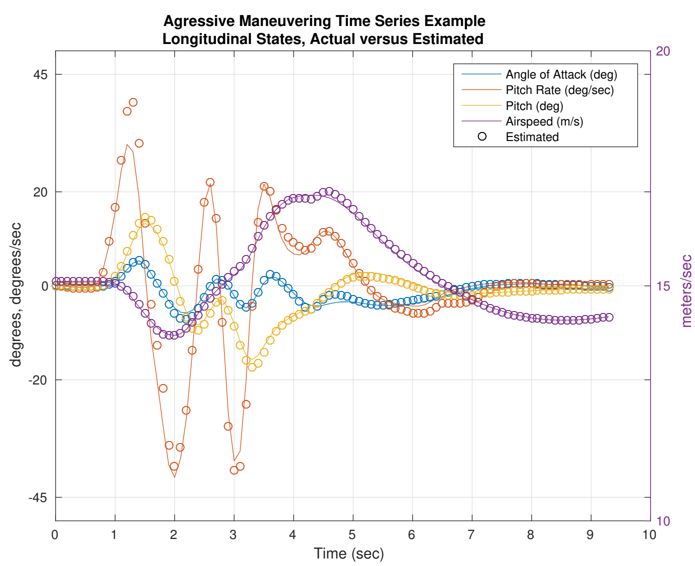

Figure 2. Time Series Comparison of Test versus Predicted States

and random number seeds. When highly-weighted samples were chosen first by the random number generator convergence was more predictable.

4) Ordered-Gradient Descent: With this technique, the same gradient descent step was performed, but the highest-weighted training sample was used first. The algorithm stepped through each remaining sample in descending order until completion. Because many training samples were used, it was not necessary to revisit a single sample more than once. Iteration was halted when the Frobenius norm of the step in the regression coefficient matrix,

$$\left\|\theta^{(t+1)} - \theta^{(t)}\right\|_F < \text{tol} \mid \text{tol} \sim 10^{-6} \tag{13}$$

or after 10,000 iterations. This algorithm generally converged within 1000 iterations for training examples.

This technique is similar in context to the numerical technique of throwing out low-weighted training samples, except that the higher-weighted ones are used sequentially instead of in a batch. In general, this approach worked a lot better in terms of robustness, and required a very small $\alpha \approx 10^{-10}$.

V. Future Work

As a next step in the machine learning pipeline, analysis could be performed on the calculated parametric models θ across the flight envelope and selected states. For example, by sweeping airspeed and aerodynamic surface states one could extract quantiative control effectiveness estimates. Implementing Locally Weighted Linear Regression to leverage parallel computing would allow use of very large datasets that could cover wide ranging corners of an aircraft's flight envelope. Finally, running the algorithm on simulated and actual flight data that includes sensor error and noise is a next step to verify robustness of this regression technique.

VI. Conclusion

A non-parametric aircraft dynamic model was created from simulated flight data using Locally Weighted Linear Regression. Time-series data of 37 vehicle states from a high fidelity simulator was formulated into a discrete Markov Decision Process framework, with the objective of predicting the next state based on the current state. The Locally Weighted Linear Regression method was modified to remove particular features from the weights and regressors. In addition, L_2 regularization was used to aid in singularity issues of the normal equation solution, however was found to have no impact on overall error. Attempts at simplifying the regression problem by feature removal did not prove to lower error, demonstrating that the method is more accurate when given a larger, more comprehensive feature set. Overall, locally weighted linear regression provided reasonable one-step state predictions. Stochastic gradient descent techniques were investigated for future scaling to much larger datasets.

References

[1] Mark B Tischler. System identification methods for aircraft flight control development and validation. *Advances in Aircraft Flight Control*, pages 35–69, 1996.

[2] Eugene A Morelli and Vladislav Klein. Application of system identification to aircraft at nasa langley research center. *Journal of Aircraft*, 42(1):12–25, 2005.

[3] Vladislav Klein and Eugene A Morelli. *Aircraft system identification: theory and practice*. American Institute of Aeronautics and Astronautics Reston, Va, USA, 2006.

[4] Lennart Ljung. Perspectives on system identification. *Annual Reviews in Control*, 34(1):1–12, 2010.

[5] Morelli E. Klein, V. *Aircraft System Identification, Theory and Practice*. American Institute of Aeronautics and Astronautics, 2006.

[6] Antoniewicz R. Krambeer K. Duke, E. Derivation and definition of a linear aircraft model. *NASA Reference Publication*, (1207), 1988.

[7] Adrian Wills and Brett Ninness. System identification of linear parameter varying state-space models. *Linear parameter-varying system identification: new developments and trends*, pages 295–316.

[8] Jay M Brandon and Eugene A Morelli. Real-time onboard global nonlinear aerodynamic modeling from flight data. *Journal of Aircraft*, pages 1–37, 2016.

[9] Gianluigi Pillonetto, Francesco Dinuzzo, Tianshi Chen, Giuseppe De Nicolao, and Lennart Ljung. Kernel methods in system identification, machine learning and function estimation: A survey. *Automatica*, 50(3):657–682, 2014.

[10] William S Cleveland. Robust locally weighted regression and smoothing scatterplots. *Journal of the American statistical association*, 74(368):829–836, 1979.