

# Mining Sequence Patterns from Wind Tunnel Experimental Data for Flight Control

Zhenyu Liu<sup>†</sup>, Wesley W. Chu<sup>†</sup>, Adam Huang<sup>‡</sup>, Chris Folk<sup>‡</sup>, Chih-Ming Ho<sup>‡</sup>

University of California, Los Angeles  
Los Angeles, CA90095, USA

{viciu, wwc}@cs.ucla.edu, {pohao, chrisf, chihming}@ucla.edu

**Abstract.** This paper presents a sequence pattern mining technique to mine data generated from a wind tunnel experiment. The goal is to discover the nonlinear input-output relationship for a delta wing aircraft. In contrast to categorical datasets, the output variable(s) in this dataset is continuous and takes distinct values, which is common in physical experiments. Directly applying existing decision tree or rule induction mining methods fails to discover sufficient knowledge. Therefore, we propose to extend current techniques by constructing sequence patterns that represent the output variations in certain ranges of selective inputs. Similar sequence patterns are clustered based on a weighted variance measure. Rules can then be derived from similar sequences to predict the output. This technique has been applied to the experimental data and generates rules useful for flight control.

## 1 Introduction

Existing data mining methods such as decision tree induction [11], rule derivation [1] or Bayesian learning [3], have largely focused on datasets with nonnumeric or categorical variables. Therefore, these methods are suitable for such applications as product forecasting or cross-selling where categorical variables prevail. However, data generated from scientific experiments are different from conventional datasets in the following aspects:

- Numerical variables involved are continuous and may take distinct real numbers within valid ranges.
- Strong casual relationship exists among these numerical variables. The outcome of one output variable is often correlated with all the input variables.

Therefore, new approaches are required to discover knowledge from these experimental data.

In this paper, we are focused on a dataset generated by the MEMS UAV (Uninhabited Aerial Vehicle) project in the Mechanical and Aerospace Engineering Department at UCLA. The data is collected from a wind tunnel in which a delta wing aircraft model is mounted [7]. Each tuple correlates one particular input configuration

---

<sup>†</sup> Computer Science Department

<sup>‡</sup> Mechanical and Aerospace Engineering Department

of the aircraft model with the corresponding force loading outputs. The goal of mining this dataset is to derive the highly nonlinear input-output relationship for the aircraft. Such knowledge will be useful for flight control. The preliminary dataset contains 192 tuples summarized from the wind tunnel experiments and provides insights into aircraft maneuvering via MEMS devices.

Traditional mining methods generate knowledge to predict output variables base on a subset of input variables. This approach is acceptable in many business-related applications where a portion of the inputs is sufficient to predict the output. In physical system control such as the delta wing aircraft with MEMS actuators, however, output variables (e.g. force and moments) are highly dependent on all input variables (angle of attack, stream velocity, actuation position, see Figure 1 and Figure 2). Under such environments, existing algorithms are unable to derive input-output relationship that covers all the cases.

To remedy this problem, we transform the original dataset by merging the output with several inputs into a composite output variable called sequence. More precisely, a sequence is defined as the output variation in a certain range of selected inputs. The transformed results enable us to cluster similar sequences via a bottom-up algorithm. Existing methods, e.g. rule induction, can then be applied to these sequence clusters. Using such an approach, we are able to derive fairly complete input-output relationship for the wind tunnel experimental data.

Scientific discovery research has been existing for more a decade. Its goal is to find knowledge that is novel, interesting, plausible, and understandable [14]. From this general perspective, scientific discovery shares common characteristics with that of knowledge discovery (data mining) in business applications. This work is strongly influenced by the scientific discovery viewpoint and yet leveraged on the existing data mining techniques in discovering interesting patterns from a scientific dataset. The resulting rules are special cases of the qualitative and quantitative laws in the general scientific discovery framework [8].

The rest of the paper is organized as follows. Section 2 gives a brief background on the aircraft control principles and shows the deficiencies of directly applying traditional methods. In Section 3, we propose the sequence clustering technique and apply it to the wind tunnel experimental dataset. Section 4 concludes the paper and provides future research directions.

## **2 Control of a Delta Wing Aircraft**

MEMS UAV (Uninhabited Aerial Vehicle), an ongoing project at UCLA, has demonstrated the possibility of using MEMS micron-scale actuation devices to control macro-scale machines, e.g., an aircraft. Such a design has numerous advantages in reducing weight, overall power consumption and radar cross-section. The project uses the “vortex” control method to provide forces and moments for controlling the aircraft. Typically, a delta wing aircraft will produce pairs of vortices above the wings (Figure 1). These vortices are sources of low-pressure flows that provide “suction”, which produces a portion of lift for the aircraft. Airflow blows toward the delta wing, first hitting the lower surface and then moving up toward the

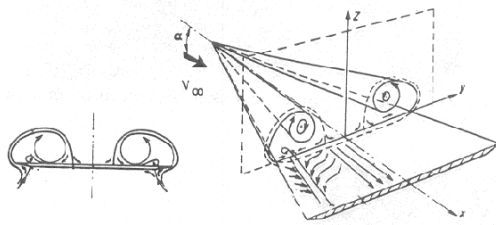


Fig. 1. Pairs of vortices above the delta wing

upper surface, eventually detaching near the leading edge and creating the vortices pair. Numerous researches have shown that the genesis location of these vortices, i.e. the detaching positions, is very important to the characteristics of the resulting large primary vortices [9, 10]. By placing MEMS actuators near this location, the symmetry of these

high “suction” vortices is broken. As a result, aerodynamic loadings on the aircraft can be controlled.

## 2.1 Problem description

The key for aircraft control is accurately predicting the aircraft’s force loadings based on certain environment settings and an actuation position. Load measurements for the delta wing, with a six-component force balance, are divided into two categories: forces and moments. Each category has three variables, corresponding to the three dimensions. Environment settings include the wind tunnel stream velocity and the aircraft’s angle of attack. Figure 2 visually interprets these terms. As shown in the figure, the delta wing is equipped with rounded leading edges. Note that the actuation position is one point at each cross section, forming a straight line along the whole leading edge. This position is represented by an angle value, ranging from  $0^\circ$  to  $180^\circ$ .

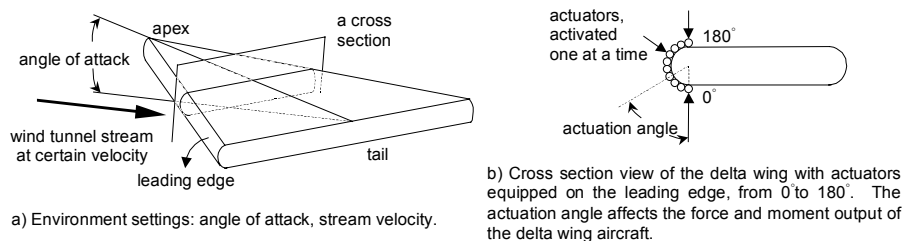


Fig. 2. Input variables for a delta wing aircraft

In the rest of the paper, the variables about the environment settings and the actuation angle are referred to as ‘input variables’, whereas the variables about the force balance outputs as ‘output variables’.

Wind tunnel experimental results have shown drastic variances of the force balance outputs with different environment and actuation settings. Table 1 shows part of the dataset that relates the rolling moment output (one component in the force balance outputs) with corresponding input values.

**Table 1.** Wind tunnel experiment results of the delta wing aircraft. Angle of attack, stream velocity and actuation angle are the input variables and rolling moment output is the output variable.

Angle of attack (°)	Stream velocity (m/s)	Actuation angle (°)	Rolling moment output
...			
20	10	40	-0.00485
20	10	60	0.00092
20	10	80	0.00026
20	10	100	-0.00621
20	10	120	0.00011
20	10	140	-0.00626
20	15	40	-0.01179
20	15	...	...
20	15	140	-0.00361
...			

## 2.2 Data characteristics

Due to current experiment design, the input variables (i.e. angle of attack, stream velocity and actuation angle) only take a small number of distinct values. Therefore, these inputs can be treated as categorical. In contrast, the output variable has distinct values for all tuples and ranges from the set of real numbers.

Furthermore, the output variable is dependent on all the input variables. Knowing even two of the three inputs is insufficient to predict the output. For example, using two variable combinations like “angle of attack = 20 and stream velocity = 10” cannot predict the rolling moment output (Table 1). This characteristic greatly undermines the effectiveness of decision tree or rule induction methods, where the output is predicted only based on a subset of the inputs. The detailed results of decision tree and rule mining are shown in the appendix.

To solve this problem, existing methods need to be extended for this dataset. Note that predicting the output based on a set of inputs is common in many physical systems. Therefore, the technique presented in this paper is general in nature.

## 3 Discovering Rules on the Basis of Sequence Clustering

The basic idea of our technique is as follows. The output value may not be determined based on a subset of the inputs. However, the output variation in certain ranges of selected inputs may follow certain sequence patterns. We shall first extract such sequence patterns from the raw data. A sequence clustering hierarchy can be built in a bottom-up fashion based on inter-cluster errors (*ice*). Such a hierarchy provides cluster candidates. A weighted variance (*wvar*) measure is used to describe the sequences closeness within each candidate. The clustering is finalized by selecting candidate clusters whose *wvars* are below a user-specified threshold. Sequences in such clusters are considered similar and approximated by the

corresponding sequence mean. Rules can be then derived on each cluster to represent the input-output relationship.

### 3.1 Definition of a Sequence

Consider a dataset  $D$  with input variables  $X_1, \dots, X_n$ , an output variable  $Y$  and a predicate  $p$  defined on the inputs. A *sequence of  $Y$  w.r.t  $X_i$  ( $1 \leq i \leq n$ ) characterized by  $p$*  is a set of 2-tuples:  $\{ \langle y_1, x_{i_1} \rangle, \langle y_2, x_{i_2} \rangle, \dots, \langle y_m, x_{i_m} \rangle \}$  calculated by  $\prod_{Y, X_i} (\sigma_p(D))$ <sup>1</sup>, where  $y_1, \dots, y_m, x_{i_1}, \dots, x_{i_m}$  are specific values of  $Y$  and  $X_i$ , respectively. Without losing generality, we can assume  $x_{i_1} \leq x_{i_2} \leq \dots \leq x_{i_m}$ . For example, Table 1 contains a sequence of the rolling moment w.r.t the actuation angle:  $\{ \langle -0.00485, 40 \rangle, \langle 0.00092, 60 \rangle, \langle 0.00026, 80 \rangle, \langle -0.00621, 100 \rangle, \langle 0.00011, 120 \rangle, \langle -0.00626, 140 \rangle \}$  characterized by “angle of attack = 20 AND stream velocity = 10”.

Note that this definition is slightly different from those in existing research, e.g., [12, 13], where the time variable is implicitly used as  $X_i$  in our definition.

### 3.2 Clustering Hierarchy Generation

Since sequences are objects with no total order, we use a bottom-up clustering strategy, MDC [15], to build a hierarchical cluster over the sequence set. Given  $s$  sequences,  $s$  initial clusters are built each containing one sequence. The algorithm merges two closest clusters at each step and finishes constructing a binary-tree after the  $s-1$ th iteration.

Let us now apply this clustering strategy to the experimental data:

1. Extract sequences of the rolling moment output w.r.t the actuation angle (Table 1). Such sequences are characterized by predicates in the form: “angle of attack =  $\alpha$  AND stream velocity =  $v$ ”, where  $\alpha$  and  $v$  range from  $\{5^\circ, 10^\circ, 15^\circ, 20^\circ, 25^\circ, 30^\circ, 35^\circ\}$  and  $\{10 \text{ m/s}, 15 \text{ m/s}, 20 \text{ m/s}\}$ , respectively.
2. In order to discover more frequently occurred patterns from these sequences, we normalize on the output variable so that for a particular angle of attack and stream velocity the difference between the maximum and the minimum output is 1.
3. Euclidean distance is used as the distance measure between two sequences  $S_i$  and  $S_j$

$$|S_i - S_j| = \left[ \sum_{k=1}^m (s_{i_k} - s_{j_k})^2 \right]^{1/2} \quad (1)$$

Here  $m$  is the length of each sequence, while  $s_{i_k}, s_{j_k}$  ( $1 \leq k \leq m$ ) are the  $k$ th output values in sequence  $S_i$  and  $S_j$ , respectively.

4. An inter-cluster error (*ice*) measure [15] is used to calculate the distance between two sequence clusters  $C_1$  and  $C_2$  ( $|C|$  denotes the size of  $C$ ):

$$ice(C_1, C_2) = \frac{1}{|C_1|} \frac{1}{|C_2|} \sum_{i=1}^{|C_1|} \sum_{j=1}^{|C_2|} |S_i - S_j|, S_i \in C_1, S_j \in C_2 \quad (2)$$

<sup>1</sup> Here  $\sigma$  is the relation selection operator, while  $\prod$  is the relation projection operator.

The resulting clustering hierarchy is shown in Figure 3. Each leaf is a sequence characterized by the corresponding label. For example, the leaf “angle of attack = 20° AND stream velocity = 10” represents the sequence  $\{<-0.00485, 40>, <0.00092, 60>, <0.00026, 80>, <-0.00621, 100>, <0.00011, 120>, <-0.00626, 140>\}$ .

### 3.3 Cluster Selection Based on Weighted Variance

Each branch node in the generated hierarchy represents a candidate cluster. We shall introduce the notion of weighted variance (*wvar*) to measure the closeness within a candidate cluster. Candidates with *wvars* lower than a specified threshold will be chosen as final clusters. For a cluster  $C = \{S_1, S_2, \dots, S_l\}$ , *wvar*(*C*) should be proportional to the cluster’s standard deviation:

$$\sigma_s = \left[ \frac{1}{l} \sum_{i=1}^l |S_i - E_s|^2 \right]^{1/2}, \text{ where } E_s \text{ is the mean of } S_1, \dots, S_l. \quad (3)$$

For two clusters with the same standard deviation but different amplitude ranges, we introduce the amplitude measure *amp*(*C*) to provide weighted preference based on a cluster *C*’s amplitude range:

$$amp(C) = \max_{1 \leq i \leq l} (\max_{1 \leq k \leq m} s_{i_k}) - \min_{1 \leq i \leq l} (\min_{1 \leq k \leq m} s_{i_k}), \quad (4)$$

where *m* is the length of each sequence and  $s_{i_k}$  ( $1 \leq i \leq l, 1 \leq k \leq m$ ) is the  $k_{th}$  output value in sequence  $S_i$ .

Therefore, we define the weighted variance for a cluster *C* as:

$$wvar(C) = \frac{\sigma_s}{amp(C)}, \text{ if } amp(C) \neq 0; \text{ otherwise } 0 \quad (5)$$

Each branch node in Figure 3 represents a candidate cluster and is labeled by that cluster’s *wvar*. All the sequences in a branch with *wvar* below certain user-specified threshold are considered similar. A smaller *wvar* threshold yields smaller cluster sizes and more accurate approximation by the sequence mean. By setting such a threshold as “*wvar* < 0.32”, the final clusters are chosen as Figure 4.

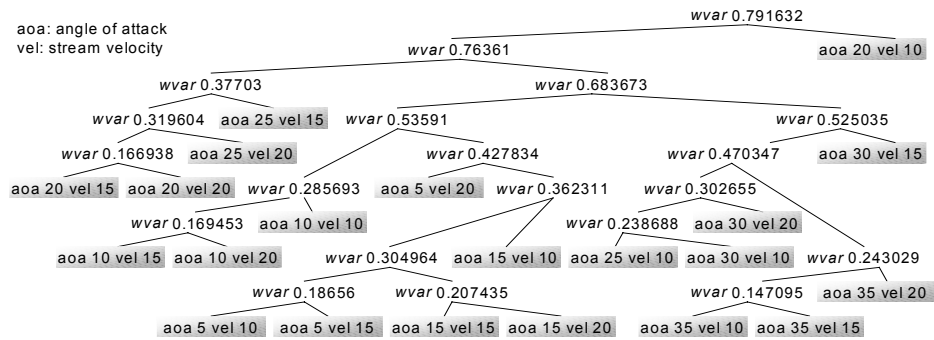


Fig. 3. Clustering of sequences extracted from the experimental flight data

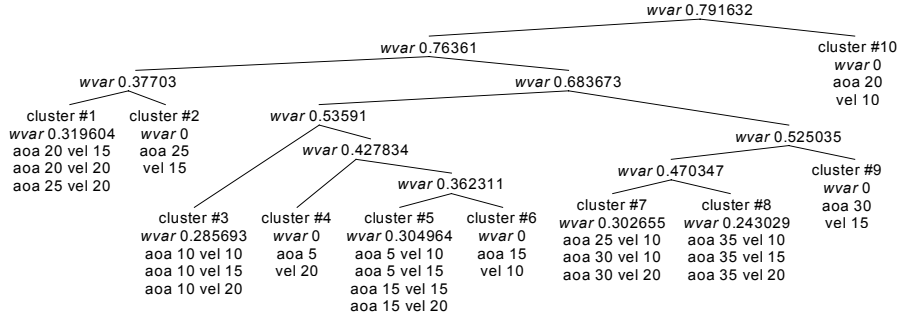


Fig. 4. The pruned clustering hierarchy of Figure 3 with  $cae < 0.32$

Based on the results in Figure 4, traditional mining methods such as rule induction can be applied to generate more complete knowledge.

### 3.4 Rule Derivation from Similar Sequences

Based on the clustering result, forward inference rules (also referred to as classification or discriminant rules [5, 6]) can be derived in the following form:

$$\text{IF } p \text{ THEN } \text{mean}(\text{cluster } \#i), \text{wvar}(\text{cluster } \#i), \text{confidence: } P[\text{cluster } \#i | p] \quad (6)$$

Here  $p$  is a predicate defined on the input variables,  $\text{mean}(\text{cluster } \#i)$  is the sequence mean calculated on cluster  $\#i$ ,  $\text{wvar}(\text{cluster } \#i)$  is the cluster's weighted variance and  $P[\text{cluster } \#i | p]$  is the conditional probability of cluster  $\#i$  given  $p$ .

To derive such forward inference rules, an algorithm should search over all possible input variable predicates and select those predicates that yield rule supports and confidences above certain thresholds. Pruning strategies are used in this process to reduce the search space. For efficient algorithms on forward inference rule generation, see [4, 5].

For example, to generate rules on cluster #5 (Figure 4), we set the minimum support as "2" and minimum confidence as "60%". The forward inference rules generated are:

- IF angle of attack=5° THEN  $\text{mean}(\text{cluster } \#5)$ ,  $\text{wvar } 0.304964$ , confidence 66.7%.
- IF angle of attack=15° THEN  $\text{mean}(\text{cluster } \#5)$ ,  $\text{wvar } 0.304964$ , confidence 66.7%

Figure 5(a) displays the rolling moment output values of the four sequences in cluster #5. Figure 5(b) shows the corresponding sequence mean.

Similarly, the following rule can be generated from cluster #8:

- IF angle of attack=35° THEN  $\text{mean}(\text{cluster } \#8)$ ,  $\text{wvar } 0.243029$ , confidence 100%.

The sequences and mean of cluster #8 are shown in Figure 6(a) and Figure 6(b), respectively.

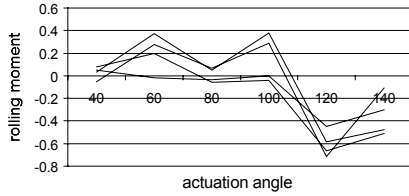
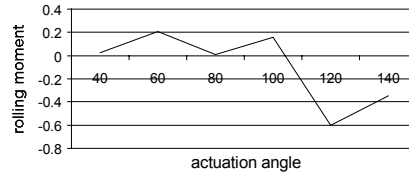


Fig. 5. (a) four sequences in cluster #5



(b)  $mean(\text{cluster } \#5)$

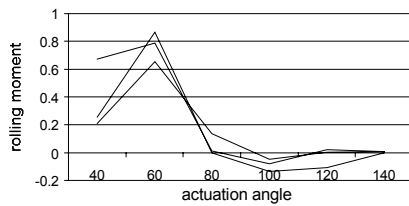
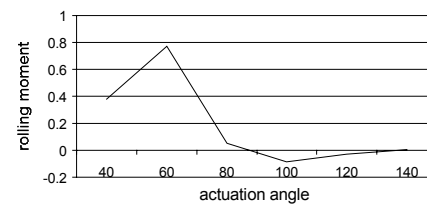


Fig. 6. (a) three sequences in cluster #8



(b)  $mean(\text{cluster } \#8)$

### 3.5 Application of Derived Rules

Since the clustering result summarizes the raw data, we can derive rules on all the clusters. The resulting rule set gives much better coverage over the entire case space, and therefore is more useful for flight control.

Currently, forward inference rules can be used to predict force balance outputs based on input settings. An average sequence is first predicted using angle of attack and/or stream velocity. For example, given an angle of attack at  $35^\circ$ , the sequence in Figure 6(b) is selected. Using this sequence, the rolling moment output can be determined at each actuation angle. Note that the sequences have been normalized before clustering (see Sect. 3.2). The output value should be multiplied by the corresponding normalization factor.

We are planning a series of wind tunnel experiments to learn the dynamic characteristics of the delta wing aircraft. Not only the force balance outputs but also the aircraft's moving dynamics (e.g. speed and acceleration) will be recorded. The augmented dataset will allow us to derive rules providing more insight into flight dynamics. For example, rules can be generated predicting the variation of the rolling speed with respect to the actuation setting and/or the roll angle. Such rules can guide us to select the proper actuation schema to achieve a desirable control effect.

## 4 Conclusion and future work

Traditional mining methods fail to derive sufficient input-output relationship for predicting physical system behavior. In this paper, we propose a novel knowledge discovery technique based on sequence patterns. A sequence is defined as the output



variation in certain ranges of selected inputs. A sequence clustering hierarchy can be built in a bottom-up fashion, using inter-cluster error (*ice*) as the distance measure. Based on the hierarchy, similar sequences are grouped in to clusters. The sizes of these clusters are controlled by the weighted variance (*wvar*) measure. Each cluster is represented by the sequence mean of that cluster. Forward inference rules are then derived on each cluster to represent the input-output relationship. We have applied this technique to the wind tunnel experimental data and derive useful knowledge for MEMS-based aircraft control.

From the experiment design aspect, we plan to expand the current wind tunnel experiments to include dynamic behaviors. Such experimental data will allow us to mine input-output relationship under dynamic environments. From the algorithm development aspect, the current sequence definition needs to be extended to include multiple input and output variables, which will widen the scope of frequent patterns. Further, we need to extend the proposed sequence clustering and rule derivation technique to future augmented datasets and reduce the computation complexity.

## Reference:

1. R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A.I. Verkamo. Fast discovery of association rules. In U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth and R. Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, pages 399-421. AAAI/MIT Press, 1996.
2. W.W. Chu and K. Chiang. Abstraction of high level concepts from numerical values in databases. In *Proc. of the AAAI Workshop on Knowledge Discovery in Databases*, Seattle, MA, July 1994.
3. C. Elkan. Boosting and naive bayesian learning. Technical report no cs97-557. Dept. of Computer Science and Engineering, UCSD, September 1997.
4. G. Giuffrida, L. G. Cooper, and W. W. Chu. A scalable bottom-up data mining algorithm for relational databases. In *10<sup>th</sup> Int'l Conf. on Scientific and Statistical Database Management (SSDBM '98)*, Capri, Italy, July 1998.
5. G. Giuffrida, W. W. Chu, and D. M. Hanssens. NOAH: An algorithm for mining classification rules from datasets with large attribute space. In *Proc. of 12<sup>th</sup> Int'l Conf. on Extending Database (EDBT)*, Konsta, Germany, March 2000.
6. J. Han and Y. Fu. Exploration of the power of attribute-oriented induction in data mining. In U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth and R. Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, pages 399-421. AAAI/MIT Press, 1996.
7. A. Huang, C. Folk, C. Silva, B. Christensen, Y. Chen, C.M. Ho, F. Jiang, C. Grosjean, Y.C. Tai, G.B. Lee, M. Chen, S. Newbern. Applications of MEMS devices to delta wing aircraft: from concept development to transonic flight test. In *39<sup>th</sup> AIAA Aerospace Sciences Meeting and Exhibit*, Reno, U.S.A., January, 2001.
8. P. Langley. The computer-aided discovery of scientific knowledge. In *Proc. of the 1<sup>st</sup> Int'l Conf. on Discovery Science (DS'98)*, Fukuoka, Japan, December 1998.
9. G.B. Lee, S. Chiang, Y.C. Tai, T. Tsao, and C.M. Ho, Robust vortex control of a delta wing using distributed MEMS actuators, accepted, *Journal of Aircraft* (2000).
10. J.F. Marchman III, "Effect of heating on leading edge vortices in subsonic flow," *Journal of Aircraft*, Vol. 12, No. 2, pp. 121-123 (1975).

11. M. Mehta, R. Agrawal, and J. Rissanen. SLIQ: A fast scalable classifier for data mining. In *Proc. of the 5<sup>th</sup> Int'l Conf. on Extending Database Technology (EDBT)*, Avignon, France, March 1996.
12. S. Park, W.W. Chu, J. Yoon, and C. Hsu. Efficient searches for similar subsequences of different lengths in sequence databases. In *Proc. of the 16<sup>th</sup> IEEE Int'l Conf. on Data Engineering (ICDE)*, San Diego, CA, February, 2000.
13. R. Srikant and R. Agrawal. Mining sequential patterns: generalizations and performance improvements. In *Proc. of the 5<sup>th</sup> Int'l Conf. on Extending Database Technology (EDBT)*, Avignon, France, March 1996.
14. R.E. Valdés-Pérez. Knowledge discovery tools for science applications. In *Communications of the ACM*, 42(11):37-41, November 1999.
15. G. Zhang and W.W. Chu. MDC: a mixed-type data clustering algorithm. Technical report, Computer Science Department, UCLA.

## Acknowledgement

The authors would like to thank Giovanni Giuffrida and Qinghua Zou for many fruitful discussions and valuable comments on this paper. The wind tunnel experiments discussed in this paper are supported by DARPA MTO.

## Appendix

Conventional data mining research mostly focused to datasets with only categorical variables. To apply existing methods on this particular dataset, we need to first discretize the output variable that takes continuous values. Such discretization methods are discussed in [6]. The basic idea is to leverage on a concept hierarchy generated manually by experts or automatically from the data distribution. The Distance Sensitive Clustering (DISC) method is used [2] to build a Type Abstraction Hierarchy (TAH) for the output variable. Each node in such a hierarchy corresponds to a value range. The whole data range can be partitioned using the ranges of nodes at a certain level. For example, the rolling moment output data in Table 1 can be partitioned into six clusters: [-0.01179, -0.00380], [-0.00380, -0.00138], [-0.00138, -0.00024], [-0.00024, 0.00028], [0.00028, 0.00124], [0.00124, 0.00537].

In the following sections, we apply two common methods to classify the rolling moment output: decision tree and association rule induction.

### Decision Tree

The dataset was first run on a decision tree generation algorithm provided by IBM Intelligent Miner [11]. The class variable is the discretized rolling moment output. The input variables include angle of attack, stream velocity and actuation angle. The tool generated a four-level decision tree after pruning. To test the effectiveness of the result, the tree was directly applied back to predict the training dataset. Table 2 shows

the prediction confusion matrix. The number in the  $(i_{th}, j_{th})$  table entry represents the percentage of tuples that belongs to the  $i_{th}$  class yet predicted as the  $j_{th}$  class.

The high error rate attributes to the high dependency of the output variable on all the input variables. A single input variable has low predictive power on the output when taken alone. Therefore, univariate splitting, the basic philosophy behind decision tree induction, makes the method unsuitable for this kind of dataset.

**Table 2.** Confusion matrix based on the pruned decision tree. Error rate: 46.35%

Predicted Classes \ Actual Classes	[-0.01179, -0.00380]	[-0.00380, -0.00138]	[-0.00138, -0.00024]	[-0.00024, 0.00028]	[0.00028, 0.00124]	[0.00124, 0.00537]	Total
[-0.01179, -0.00380]	6.25%	0%	0%	1.56%	0%	1.04%	8.85%
[-0.00380, -0.00138]	1.56%	0%	0%	6.77%	0%	1.56%	9.90%
[-0.00138, -0.00024]	2.60%	0%	0%	10.42%	0%	2.60%	15.62%
[-0.00024, 0.00028]	3.02%	0%	0%	38.54%	0%	0.52%	42.18%
[0.00028, 0.00124]	4.17%	0%	0%	6.25%	0%	1.04%	11.46%
[0.00124, 0.00537]	1.04%	0%	0%	2.08%	0%	8.85%	11.98%
Total	18.75%	0%	0%	65.62%	0%	15.62%	100%

### Association Rules

Our second effort was to run the dataset on association rule derivation methods. Since we are concerned about using input variables to predict the output variable, we concentrate only on rules that have input variables in their left-hand-sides and the output variable as their right-hand-sides. The Apriori algorithm [1] has been tested on the dataset after discretization. The minimum support and confidence are set to 3% and 70%, respectively. All the resulting rules that satisfy the above restriction are listed in table 3.

**Table 3.** Rules generated by Apriori associate the output variable with the input variables

#	Support (%)	Confidence (%)	Rule body	Rule head i.e. the rolling moment
1	11.9792	95.8300	angle of attack = 0	[-0.00024, 0.00028]
2	4.1667	100.0000	angle of attack = 0 AND stream velocity = 10	[-0.00024, 0.00028]
3	3.6458	87.5000	angle of attack = 0 AND stream velocity = 15	[-0.00024, 0.00028]
4	4.1667	100.0000	angle of attack = 0 AND stream velocity = 20	[-0.00024, 0.00028]
5	9.8958	79.1700	angle of attack = 5	[-0.00024, 0.00028]
6	3.1250	75.0000	angle of attack = 5 AND stream velocity = 10	[-0.00024, 0.00028]
7	3.1250	75.0000	angle of attack = 5 AND stream velocity = 15	[-0.00024, 0.00028]
8	3.6458	87.5000	angle of attack = 5 AND stream velocity = 20	[-0.00024, 0.00028]
9	3.1250	75.0000	stream velocity = 10.00 AND actuation angle = 0	[-0.00024, 0.00028]

The knowledge provided by those rules suffer from the following shortcomings:

1. **Low coverage.** The nine rules in Table 3 cover only 28.125% of the original dataset, whereas 71.875% of the cases encountered cannot be predicted. Due to the low coverage over the entire case space, this rule set cannot provide sufficient information about the input-output relationship. Thus, it is insufficient for flight control.

2. **Unable to handle control-sensitive regions.** When the angle of attack is above 15°, the output variable is more sensitive to the inputs. That is, the output in this region has larger magnitudes and greater variances. However, the rules (Table 3) derived by Apriori are mostly in the insensitive region (i.e. angle of attack below 15°) since data in this region is less variant and tends to give higher rule supports and confidences.

The reason that association rules fail to capture the sensitive region is due to the basic rule form: “ IF  $X_1 = x_1$  AND ... AND  $X_t = x_t$  THEN  $Y = y$ ”. Here  $X_1, \dots, X_t$  are input variables and  $Y$  is the output. For a dataset with  $n$  input variables,  $t$  is usually less than  $n$ . Otherwise a rule simply reiterates a tuple in the dataset. However, a  $t$  less than  $n$  means omitting certain input variables. In the sensitive region, omitting any input variable in the left-hand-side may be disastrous since the right-hand-side cannot be concentrated in one category. This is best illustrated by the real data shown below.

**Table 4.** Dropping any one variable in “angle of attack = 20° AND stream velocity = 15 AND actuation angle = 60°” generates three 2-variable combinations. Each of these 2-variable combinations corresponds to a sub table listed below. From these tables, we note that no rules containing only two variables in the left-hand-side, e.g. “IF angle of attack = 20° AND stream velocity = 15 THEN Rolling moment = ...”, will have a big confidence measure.

Angle of attack (°)	Stream velocity (m/s)	Actuation angle (°)	Rolling moment
20	15	40	[-0.01179, -0.00380]
20	15	60	[-0.00380, -0.00138]
20	15	80	[-0.01179, -0.00380]
20	15	100	[0.00124, 0.00537]
20	15	120	[0.00028, 0.00124]
20	15	140	[-0.00380, -0.00138]

a) angle of attack = 20 AND stream velocity = 15

Angle of attack (°)	Stream velocity (m/s)	Actuation angle (°)	Rolling moment
20	10	60	[0.00028, 0.00124]
20	15	60	[-0.00380, -0.00138]
20	20	60	[-0.01179, -0.00380]

c) angle of attack = 20 AND actuation angle = 60

Angle of attack (°)	Stream velocity (m/s)	Actuation angle (°)	Rolling moment
0	15	60	[0.00028, 0.00124]
5	15	60	[-0.00024, 0.00028]
10	15	60	[-0.00024, 0.00028]
15	15	60	[0.00124, 0.00537]
20	15	60	[-0.00380, -0.00138]
25	15	60	[0.00028, 0.00124]
30	15	60	[0.00124, 0.00537]
35	15	60	[0.00124, 0.00537]

b) stream velocity = 15 AND actuation angle = 60

Thus, using conventional rule induction results in the following dilemma: rules generated either reiterate the original tuples, or have undesirably low confidences.