Higher State Predictions of Coaxial Rotor Helicopters Using XGBoost Gradient Boosted Trees

Ethan Genter  
*Washington University in St. Louis*

Cory Seidel  
*Washington University in St. Louis*

David A. Peters  
*Washington University in St. Louis*

Follow this and additional works at: [https://openscholarship.wustl.edu/mems500](https://openscholarship.wustl.edu/mems500)

**Recommended Citation**  
[https://openscholarship.wustl.edu/mems500/142](https://openscholarship.wustl.edu/mems500/142)
Higher State Predictions of Coaxial Rotor Helicopters Using XGBoost™ Gradient Boosted Trees

Ethan Genter
e.genter@wustl.edu
Master’s Student
Washington University in St. Louis
St. Louis, MO, USA

Cory Seidel
cory.seidel@wustl.edu
Adjunct Professor
Washington University in St. Louis
St. Louis, MO, USA

David Peters
dap@wustl.edu
Professor
Washington University in St. Louis
St. Louis, MO, USA

ABSTRACT

In this paper, the use of XGBoost™ gradient boosted trees for the prediction of on-disk velocity in a coaxial rotor helicopter is analyzed for higher state data with extremely sparse data sets. In particular, the use of these machine learning algorithms was evaluated for their prediction capabilities when intermediary state data was both reduced and withheld. This analysis showed a distinct tradeoff between model characteristics in order to produce the best performing models, as has been consistent with previous work. Additionally, it was found that these models can perform sufficiently well to predict higher harmonic solutions across the rotor disk when only trained on lower state data and a single higher state validation set. This result indicates that application to finite-state inflow modeling, and in particular, higher harmonic solutions, could help to significantly reduce the associated computational cost of higher harmonic solutions.

INTRODUCTION

Finite-state inflow models are critical to the development and understanding of the foundational physics of the helicopter dynamics that construct flight simulators. These foundations serve as the ground truth, or true physics of these helicopter systems. Currently, finite state inflow models are very efficient for single-lifting rotors and coaxial rotor systems. However, the recent expansion of these models into multirotor systems (Ref. 1) has introduced a significantly higher computational cost for higher harmonic solutions \((M > 9)\). This increase in computational cost, and therefore computation time, limits the real-time analysis capabilities of finite state methods. This increase in computational cost is largely driven by the need for a higher number of states in higher harmonic solutions, which both introduces more complex dynamics and increases the number of equations necessary to solve the system. The increasing dynamic complexity of higher harmonic systems can be seen in Fig. 1 below which shows the velocity on the lower rotor of a coaxial rotor system for an increasing number of states. This work explores the use of gradient boosted trees to accurately predict the velocity on the lower rotor of a coaxial rotor system at a higher number of states while significantly reducing training data, specifically training data at intermediate states.

While several publications have explored the use of machine learning in the field of vertical flight, as of yet there has been no investigation into the reduction of intermediate state data for prediction of higher state velocity data. Relevant publications to this work include the work of Koppejan (Ref. 2), Bagnell (Ref. 3), and Dracopoulos (Ref. 4) in developing machine learning algorithms for applications in autonomous helicopter flight. In particular, the study of autonomous flight through reinforcement learning has focused on applications to aerobatic maneuvers (Ref. 5-6). Each of these investigations has expanded the scope of machine learning application in vertical flight. However, none of these investigations focused on the use of gradient boosted trees in drastically reduced datasets, nor have any of these publications explored the prediction of higher state data without the use of significant preceding state training data.

The research for this abstract seeks to build upon the prior work of Seidel and Genter (Ref. 7-8). Each of the prior publications has explored the use of XGBoost and gradient boosted trees for their application of predicting on-disk velocity in coaxial rotor helicopters. The first in this line of research (Ref. 7) served as a true proof of concept, exploring the foundational capabilities of gradient boosted trees and the general trends that arose in hyperparameter tuning and fitting testing data to actual data. Ref. 8 extended this proof of concept to lower density datasets to understand the patterns and tradeoff of models trained on limited datasets. Each of these publications was ultimately aimed at developing a machine learning algorithm that could help mitigate the computational cost of higher harmonic finite-state inflow simulation of coaxial rotor systems. The next paper in this line of research, recently submitted to the Journal of Aircraft, therefore sought to predict higher state datasets using limited data and translate the techniques used in previous publications. This paper furthers previous work aimed at predicting higher state sets, though it diverges from all previous work in its method and data reduction. The presented work establishes a more efficient
predictive model through a significant reduction in training data which predicts higher state on-disk velocity without
the use of intermediate state data (i.e., if predicting 12 state data, models are trained on 2, 4, and 6 state data,
withholding 8 and 10 state data).

METHODOLOGY

The dataset utilized in this work is that of a coaxial rotor system in axial flow (climb) that was generated using the
model developed in Ref. 1. Here, a model with an infinite-number of blades is used, and specific focus is given to
lower rotor velocities. Lower rotor velocities were analyzed for their more interesting and more complex dynamics,
but the velocity on the upper rotor could be analyzed in a similar manner and the underlying methods should produce
similar results. The rotor parameters are varied for the nondimensional radial location on the disk (\( \bar{r} \)) and the following
system parameters: rotor spacing (d), rotor solidity (\( \sigma \)), and Lock Number (\( \gamma \)). The range of the parameters are as
follows: \( \bar{r} = 0-1 \) (step size \( \Delta \bar{r} = 0.01 \)), \( d = 0.1-2 \) (step size \( \Delta d = 0.1 \)), \( \sigma = 0.05-0.15 \) (step size \( \Delta \sigma = 0.01 \)), and \( \gamma = 5-8 \)
(step size \( \Delta \gamma = 0.5 \)). The original data set contained over 150,000 data points but was trimmed down, to varying
degrees, for different data density investigations. The goal of the work being to observe how much training data of
higher state velocities was needed to achieve an accurate model in XGBoost\textsuperscript{TM}. This work varies the number of lower
states, and therefore the amount of data, used for the training data to study profile development and denote model
performance across a predicted higher state profile. This is done both through reducing the amount of data used in
training from a lower state and by eliminating some lower states altogether. The dataset is separated into data for
training and validating the model, and the withheld data. The model is trained on all of the data from the lower number
of states and validated on a single set of simulation data from the higher state data that is desired for prediction. The
withheld data is the remaining higher state data for predicting which only exists as a form for analyzing the methods.

Within XGBoost\textsuperscript{TM}, there are several hyperparameters that adjust how a particular model’s trees are formed
and how that model fits to the data. This work utilizes variations in several hyperparameters. Most notably, the
hyperparameters that control the number of trees (NT), depth of trees (MD), and learning rate (LR), which dictates
how fast the model fits, are each evaluated. Several thousand combinations of these hyperparameters are tested to
determine the best sets for fitting the data and to make sure that the model is not underfit or overfit. In order to more
accurately tune the appropriate hyperparameters to develop the best performing model, the window of hyperparameter
variation of narrowed and discretized with smaller variations in each hyperparameter over several rounds of
evaluation. Upon refining the hyperparameter search such that there was an acceptable degree of performance in the
top performing models, models were evaluated across validation sets and against one another for trends in the fitting
and performance.

RESULTS AND DISCUSSION

The investigation presented here tests four main training-state combinations. Each of these combinations can be seen
in Table 1 below. Combination 1 used 2, 4, 6, and 8 state training data and then tested on 10 state training data.
Combination 2 scales back the training data used in combination 1 to 2 and 4 states only and again tests on 10 state
training data. Therefore, there is a significant reduction in training state data from combination 1 to 2. Combinations
3 and 4 scale testing up to 20 state data. Each combination, 3 and 4, is trained on 2, 4, 6, 8, and 10 state training data
and is then tested on 20 state training data. Again, note that each combination, 1, 2, 3, and 4, uses at least one higher
state validation set in training. The difference between combinations 3 and 4 is the number of validation sets the model
has used prior to performance evaluation. Combination 3, like combinations 1 and 2, has been validated once, while
combination 4 has been validated twice. Table 1 also shows the performance of each model in terms of L2 norm error.
Error is shown for the top performing model identified by the algorithm’s initial validation, or in the case of
combination 4, the second validation. One can see that error decreases for larger training data sets (i.e., using training
data sets with a higher number of states), as is shown in the comparison between combinations 1 and 2. Moreover, it
can be seen that error decreases with increased validation sets, as is shown in comparison of combinations 3 and 4.
However, this improvement was not observed to be uniform across variations in the number of states used in training
and testing. In particular, it was found that to accurately predict data of a lower number of states (e.g., 10 states)
validation sets beyond the first validation did not see significant improvement, while up to the second validation set
was seen to provide significant improvement for higher state prediction (e.g., 20 states).

Table 2 below shows the L2 norm error for each combination for the top 10 performers identified by the
algorithm. One can see that the model identified as the best performer does not necessarily result in the best performing
model when results are compared to actual on-disk velocity data. However, the relative performance of each model
within the top 10 as compared to one another illustrates that the variation in relative performance from one model to
another is relatively consistent with what was predicted by the algorithm. That is, if a model was predicted to be the third best performer in the top 10, the testing data corroborate the rank of this model in the top 10 to within 1 or 2 places. This gives confidence that the algorithm is picking the best performing models. This is particularly true of combinations 1. However, as training data is reduced the ability of the algorithm to identify the best performing parameter is diminished and the relative rank of different models may vary by several positions within the top 10. So, while the algorithm may capture the top 10 performing models the top performer may be identified as the tenth best performer. In some instances, the rank may stray more significantly, and the top 10 identified models may fail to capture the best performer. This is one of the tradeoffs observed as training data is significantly reduced, though it should be noted that despite this tradeoff, the L2 norm error is still relatively low and even models with a very low number of training states such as combination 2 perform well. However, when scaling these models up to 20 state prediction, this pattern is not necessarily true based on the amount of validation sets that a model has experienced. For instance, while combination 3 is able to capture the best performer within its top 10 identified models, and even places this model in the third highest position, every other model identified within the top 10 falls below (i.e. anywhere from a couple of positions to tens of positions below) the 10 models that actually performed best. This indicates a significant lack of ability to predict the best performing model. Following an additional validation set the results look much different, where combination 4, similar to combination 1, has identified most of the top 10 performers within a couple to a few positions of their actual rank.

Individual model performance for combinations 1 and 2 can be seen in Fig. 2 and Fig. 3 below. Each figure shows both the lower state training data in green, the actual higher state data in blue, and the higher state data predicted by the model in red. It is observed that combination 1 displays clear improvements to model fitting as compared to combination 2. Combination 1 fits to data better across the entirety of the rotor disk, while combination 2’s predicted values stray further from the actual data, particularly at the tip where change in velocity is most dramatic as well as at the root of the blade in the region of $\bar{r} = 0 - 0.3$. Deviation from the actual data is expected in these regions and failure of the model to accurately predict velocity at the root and tip of the blade, especially for reduced data density, has been shown in Ref. 8. However, one can see in Fig. 3 that combination 2 loses accuracy across the blade, as compared to combination 1. In the region of $\bar{r} = 0.3 - 0.6$ in Fig. 3 this discrepancy is highlighted by the clear presence of an underlying step function with sections of constant velocity data predicted in discrete blocks over a range of $\bar{r}$.

Given that data has been significantly reduced for combination 2, one can expect that a failure of the model to accurately predict velocity in these regions is likely due to either poor splitting locations, not enough splits in the model, or not enough data/variation in number of states to recognize trends. The performance of these models and identification of regions of poor performance was corroborated by direct comparison of predicted versus actual velocity. A plot of this comparison for combination 2 can be seen in Fig. 4, where the red one-to-one line represents perfect accuracy between predicted and actual velocity. One can see that velocities near -0.05 – 0 are overpredicted (i.e., above the one-to-one line), and velocities near 0 – 0.07 and 0.2 – 0.3 are underpredicted (i.e., under the one-to-one line). When compared to Fig. 3, these velocities correspond to regions near the tip and root of the blade, which was expected. It is observed that performance at both of these locations improves for combination 1, the result of additional training data at higher states that begins to hone in on the velocities at these locations. Despite diminishing accuracy seen between Fig. 2 and Fig. 3, as seen in Table 1, these discrepancies do not lead to a poor fitting model, in fact a model trained on 2 and 4 states can still be considered a good performer with relatively low L2 norm error. It is common in vertical flight applications that experimentally gathered data often includes the elimination of data from the first 10% of the blade starting at the root. This would likely further improve the model fit, though the model would also benefit from increased and refined data at the tip.

The individual model performance of the 20 state predictions, that is, combinations 3 and 4, can be seen in Fig. 5 and Fig. 6. Again, the lower state training data is shown in green, the actual higher state data in blue, and the higher state data predicted by the model in red. Combination 4 shows clear improvements to combination 3 particularly in the region of $\bar{r} = 0.5 - 0.9$. The improvement of combination 4 over combination 3 should be expected given that combination 4 has experienced a second validation, allowing this model to learn more on the same amount of training data and training data of the same number of states. While combination 4 improves the prediction of the model across nearly the entire rotor disk, it does deviate further from the performance of combination 3 in some regions, namely the tip of the blade and in the region of $\bar{r} = 0.2 - 0.3$. The comparison of these two models is made even clearer through Fig. 7 which shows a direct comparison of predicted versus actual velocity for both combination 3 and combination 4. The improvement in combination 4 is clear as almost all points migrate toward the one-to-one as compared to their positions in combination 3. The regions of poorer performance in combination 4 are further corroborated by Fig. 7 as well, with a clear deviation of points near velocities of 0 – 0.05 occupied by the blade tip and $\bar{r} = 0.2 - 0.3$. Overall, we see improvement from combination 4 to combination 3, though it should be noted that with combinations 1 and 2, both combinations 3 and 4 have models that fail to capture the dynamics of the
higher state data in several regions. While combinations 1, 2, 3, and 4 all fail to accurately model the dynamics at the tip of the blade, combinations 3 and 4 struggle less than combinations 1 and 2 do with predictions at the blade root. Instead, the decreased performance of combination 3 is observed across several regions of the rotor disk, particularly in regions of higher change in velocity. Again, the discrepancies between the two models are likely due to either poor decisions in regards to splitting, splitting locations, etc. When compared to combinations 1 and 2, it is significant that even when predicting a much higher number of states (i.e., 20 as opposed to 10), and therefore a much more complex velocity profile across the rotor disk, combination 3 still performs relatively well, and combination 4 performs very well.

The results of these simulations show that the implemented algorithm, and XGBoost™ and gradient boosted trees, has the ability to train on significantly reduced training datasets, only including data from a lower number of states (e.g., 2, 4, 6, 8 states or 2, 4, 6, 8, 10 states) and a single higher state validation set, and predict data at a higher number of states (e.g., 10 or 20 states). Effectively skipping intermediary training data sets for higher state velocities could be particularly advantageous for finite-blade models where using lower state data would much less costly than higher state data.

![Variation in velocity profile shape across the rotor with differing number of states.](image)

**Figure 1:** Variation in velocity profile shape across the rotor with differing number of states.

<table>
<thead>
<tr>
<th>Combination</th>
<th>Training States</th>
<th>L2 Norm Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2, 4, 6, 8</td>
<td>4.199</td>
</tr>
<tr>
<td>2</td>
<td>2, 4</td>
<td>4.459</td>
</tr>
<tr>
<td>3</td>
<td>2, 4, 6, 8, 10</td>
<td>5.048</td>
</tr>
<tr>
<td>4</td>
<td>2, 4, 6, 8, 10</td>
<td>3.371</td>
</tr>
</tbody>
</table>

**Table 1:** The four combinations of varied training state sets used in model performance analysis.

<table>
<thead>
<tr>
<th>Predicted Rank</th>
<th>Combination 1</th>
<th>Combination 2</th>
<th>Combination 3</th>
<th>Combination 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4.199</td>
<td>4.459</td>
<td>5.048</td>
<td>3.371</td>
</tr>
<tr>
<td>2</td>
<td>3.136</td>
<td>3.465</td>
<td>7.611</td>
<td>2.944</td>
</tr>
<tr>
<td>3</td>
<td>3.751</td>
<td>3.699</td>
<td>2.954</td>
<td>4.109</td>
</tr>
<tr>
<td>4</td>
<td>5.012</td>
<td>4.596</td>
<td>5.005</td>
<td>3.817</td>
</tr>
<tr>
<td>5</td>
<td>3.897</td>
<td>3.353</td>
<td>5.048</td>
<td>4.144</td>
</tr>
<tr>
<td>6</td>
<td>4.637</td>
<td>3.225</td>
<td>3.744</td>
<td>4.155</td>
</tr>
<tr>
<td>7</td>
<td>4.133</td>
<td>4.187</td>
<td>4.902</td>
<td>4.216</td>
</tr>
<tr>
<td>8</td>
<td>3.664</td>
<td>4.001</td>
<td>4.423</td>
<td>4.285</td>
</tr>
<tr>
<td>9</td>
<td>4.120</td>
<td>4.615</td>
<td>4.434</td>
<td>4.053</td>
</tr>
</tbody>
</table>

**Table 2:** Relative performance of the models identified as the top 10 for combinations 1, 2, 3, and 4.
Figure 2: Comparison of actual 10 state velocity profile data with training data and predicted data of combination 1.

Figure 3: Comparison of actual 10 state velocity profile data with training data and predicted data of combination 2.

Figure 4: Comparison of predicted and actual velocity for combination 2.
Figure 5: Comparison of actual 20 state velocity profile data with training data and predicted data of combination 3.

Figure 6: Comparison of actual 20 state velocity profile data with training data and predicted data of combination 4.

Figure 7: Comparison of predicted and actual velocity for combinations 3 and 4, respectively.
CONCLUSION

The ability of gradient boosted tree models to fit to variations in the amount of lower velocity state training data across the lower rotor of a coaxial rotor helicopter is presented here. The reduction of training data such that intermediary state data is not used for training is of particular focus. The key developments from this work are:

1. The use of gradient boosted trees is still a valid and accurate approach for prediction when intermediary state data has been excluded from training data sets.
2. Intermediary state data can be significantly reduced in training data sets for prediction of significantly higher state velocities (e.g. 20 states) and maintain accurate prediction capabilities.
3. For prediction of a smaller number of states (e.g. 10 states) a single validation set is sufficient to develop models of accurate performance, as well as predict some of the highest performing models. However, prediction of a higher number of states (e.g., 20 states) requires two validation sets to achieve comparable performance.
4. Models predicting a lower number of states (e.g. 10 states) struggle to predict dynamics at the tip and root of the blade, while models predicting a higher number of states (e.g. 20 states) share error in prediction across the rotor disk, although poor prediction at the blade tip is also notable.

ACKNOWLEDGMENTS

This research was partially funded by the Government under Agreement No. W911W6-17-2-0002. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the US Army Aviation Development Directorate, CCDC AvMC.

REFERENCES