# Title of Your Prospectus

by

Andrew Tagg

# BYU

## MECHANICAL ENGINEERING

Prospectus Approval

Prospectus submitted by:

_____          _____
Andrew Tagg                                          Date


This prospectus has been approved by each member of the Graduate Committee:

_____          _____
Committee Member - Chair                        Date


_____          _____
Committee Member                                     Date


_____          _____
Committee Member                                     Date

# 1    Problem Statement

High fidelity aerodynamic modeling is an essential tool in the analysis of modern aircraft, yet its use in applications such as design optimization and real time control is severely limited due to high computational costs. Instead, current optimization and control loops mainly rely on simplified, steady state aerodynamic models which are desirable for rapid evaluation, but neglect essential characteristics of unsteady flight. If we can overcome the computational hurdle, and incorperate high fidelity, unsteady aerodynamics into our design and control algorithms, we can significantly improve the performance, safety, and reliability of next-generation aircraft systems.

In this work, we propose a framework to fully incorporate high fidelity aerodynamics into optimization and control loops through an unsteady vortex particle method (VPM). We will accelerate VPM simulation with a novel deep learning technique designed to reduce the number of time steps required to solve the governing differential equations. This will significantly reduce computational cost, thereby enhancing the practicality of the VPM in large scale iterative optimization. For real time control, we will leverage Koopman operator theory to learn linear representations of VPM dynamics, and develop methods to improve the generalizability of the Koopman learning process. By learning linear representations of the underlying flow dynamics, we will support rapid and accurate control synthesis using tried and true linear techniques.

By developing these key innovations, we will enable large scale integration of VPM aerodynamics into the design and control process. High fidelity aerodynamics will inform decisions from early conceptual design all the way through to the flight control stage. These decisions will not only consider how the aircraft behaves, but also how the surrounding air moves and evolves during complex maneuvers. With these enhaced modeling capabilities available during moments of crucial decision making, we will open the door for aircraft that are not only more efficient and capable, but fundamentally more intelligent in how they interact with their environment.

# 2    Background

The next generation of aircraft systems is being conceved to operate in increasingly challenging conditions, and execute mission profiles that span multiple diverse flight regimes. Some aircraft are capable of morphing their wings to access the post stall region for precision landings [1]. Other biologically inspired vehicles replicate nature and achieve remarkable agility [2, 3]. Electric vertical takeoff and landing (eVTOL) aircraft employ lifting propulsors and fixed wing components to takeoff in hover and then transition to cruise.

All of these aircraft pose significant operational challenges, because they must undergo complex dynamic maneuvers in which unsteady aerodynamics play a leading role. For example, the eVTOL transition from hover to cruise is largely characterized by wake mixing, and unsteady interactions between propellers and lifting surfaces that are difficult to predict. In the presence of strong unsteady loading, the aircraft must execute a series of complex control inputs that balance forward motion with vertical propulsion while simultaneously managing rotation. In addition, successful transitions often require adherence to constraints on power

[4], acceleration [5], acoustic output [6], and obstacle avoidance [7]. Considering all of these factors at once quickly becomes an overwhelming task for a pilot.

The key to navigating this complex space of decisions is trajectory optimization, which is the process of optimizing the path, and control inputs of the maneuver to minimize some desired objective subject to required constraints. Recent work in eVTOL trajectory optimization has exposed key insights that highlight the importance of consdering complex aerodynamic phenomena when optimizing these transition maneuvers. [8] optimized the takeoff trajectory of a tilt-wing aircraft, and found that optimal trajectories involve operating at or near the stall region. This was complemented by other work which investigated the possibility of using active flow control to delay stall during a tilt-wing transition [9]. Another study directly implemented a constraint to avoid the vortex ring state during takeoff and landing and showed how this constraint afffected the optimal path [10].

Considering these unsteady aerodynamic phenomena during trajectory optimization is crucial for safe and optimal performance, and therefore affects overall aircraft design decisions. Recent work has shown that trajectory optimization plays a key role in battery sizing [11], and can inform airfoil design [9]. This is why several design studies have directly incorperated the trajectory into the optimization process [12, 13, 14].

One of the most important results of trajectory optimization is it's natural application to automation and control. Classical control methods often rely on linear models that are only valid locally, and cannot be simultaneously applied to the entire flight envelope of a modern transitioning vehicle. Instead, many efforts have employed trajectory optimization as a planning process to develop controllers that evolve over time through gain scheduling [15], or onboard control policy updates [16].

Notably, there are approaches that allow for the use of a single control policy for the entire transition. These either employ advanced robust control strategies such as $H^\infty$ methods [17], or rely on Koopman operator theory to learn globally linear dynamic representations [18, 19, 20]. Koopman operator theory presents an especially promising option because it provides a way to learn dynamics that are linear everywhere in the state space, thus opening the door for linear controllers that are accurate and can be applied in any flight regime without sacrificing performance. The main limitation preventing the widespread use of current Koopman methods is that they require exhaustive data to be completely general, which is difficult to obtain for high dimensional or high fidelity problems.

The large foundation of work in trajectory optimization and control of aircraft which undergo complex transition maneuvers represents a robust and versitile framework for performing optimization and control across multiple flight regimes. This existing work has exposed critical design tradeoffs for these types of aircraft, and higlighted the importance of considering time varying dynamics during the transition phase.

However, the main limitation of the current base of research is that nearly all of these studies rely heavily on overly simplified aerodynamic models, which either assume steady state flow, or other idealizations that are not accurate during an unsteady maneuver. This is a real problem because there is strong evidence to suggest that model fidelity has a severe effect on the outcome of trajectory optimization [21]. Some studies have directly compared steady state models with unsteady models during an eVTOL transition and found significant discrepancy [22]. Others have quantified the effect of unsteady aerodynamics during rotational motion, and found that it greatly impacts the dynamic stability of the

2

aircraft [23]. Clearly, unsteady aerodynamics play an important role during the transition of these aircraft, and cannot be justifiably ignored during analysis and optimization.

Some effort has been made toward improving model fidelity in optimization and control applications. One common approach is to generate high fidelity CFD data, and then tune the parameters of the low fidelity model [24, 3, 25]. This is definitely an improvement because the new parameters are informed by high fidelity simulations, but unless the parameters are updated constantly, it is still subject to the limitations of a steady state model. Other studies have focused on applying fast unsteady aerodynamic models like the harmonic balance method [26] and Theodorsen's theory [27]. While useful, these methods apply only to very simple oscillatory motions and cannot represent a more complex maneuver.

The best effort so far to fully incorporate high fidelity simulations into optimization and control loops has come from a pair of studies that employed a light weight two-dimensional vortex particle method to trajectory optimization [28] and then real time control [16] for fixed wing perching maneuvers. Even though these VPM models were limited to two dimensions, and under 100 particles, the studies showed improved performance and more accurate precision landings when the unsteady model was used. This is direct evidence that employing high fidelity models in control loops can lead to better results, but the scope of these improvements is still very much limited by computational cost as demonstrated by the need for such a light weight version of the VPM.

One promising alternative is the use of deep learning to capture unsteady fluid dynamics. There have been many studies which apply vanilla neural networks [29, 30], convolutional neural networks [31, 32], and physics informed neural networks [33, 34] to learn fluids flows or aerodynamic coefficients. Other papers have attempted to learn high resolution flow fields from low resolution solutions [35, 36], or focus on learning the underly dynamics of the system using neural ODEs [37, 38]. These approaches offer a compelling solution to the problem of high fidelity optimization and control because they allow for rapid evaluation of the deep surrogate network at significantly reduced cost. Despite the significant advancements in deep learning, existing data-driven approaches often struggle to generalize completely unless exhaustive training data are provided, which is extremely difficult in the high dimensional and computationally expensive world of CFD.

Deep learning offers the potential for rapid evaluation and can capture unsteady aerodynamic effects, but traditional data-driven approaches often fail to generalize without prohibitively large datasets. Low-fidelity models are computationally efficient and generalize well, yet they neglect critical unsteady phenomena. High-fidelity simulations, on the other hand, are accurate and broadly applicable but are too computationally expensive for practical use in iterative optimization and real-time control. What is needed is a methodology that combines speed, accuracy, and generality, enabling unsteady aerodynamics to directly inform design and control decisions.

The proposed work addresses this challenge through a novel integration of high-fidelity simulation and deep learning. Rather than using data to fully replace the vortex particle method (VPM), we focus on accelerating the simulation of known VPM dynamics by reducing the number of required time steps. This approach will make VPM-based trajectory optimization significantly more feasible by reducing a primary bottleneck in VPM simulation. Building on this foundation, we will develop a method for learning Koopman linear representations that leverages known PDE solution methods and deep learning, enhancing

3

generalizability across flow conditions. The resulting framework will enable real-time control informed by high-fidelity, unsteady aerodynamics, bridging the gap between computationally expensive simulations and practical optimization and control applications.

# 3   Research Objectives

The central goal of this work is to facilitate the incorporation of high-fidelity, unsteady aerodynamics into optimization and control processes. To accomplish this goal, we will focus on four primary research objectives.

1. **Demonstrate VPM-based trajectory optimization for an eVTOL transition maneuver.** We will apply the VPM to simulate unsteady aerodynamic effects during an eVTOL hover-to-cruise maneuver and use this model to optimize the transition trajectory. This objective will address practical challenges associated with VPM-based optimization, such as noise and convergence, and provide a baseline in terms of computation time for comparison against future VPM acceleration efforts. Included in this study will be a comparison of high fidelity and low fidelity models and an analysis of their impact on the trajectory optimization.

2. **Develop a methodology to accelerate VPM simulations by reducing the number of required time steps.** By addressing a major computational bottleneck in VPM, this objective aims to make high-fidelity simulations fast enough to be used in iterative optimization. We will validate the method by quantifying the speedup and the accuracy of the accelerated VPM simulations.

3. **Develop a methodology to learn Koopman linear representations of VPM dynamics and improve generalizability.** This objective will seek to uncover linear representations of VPM dynamics, enabling real time control synthesis informed by VPM aerodynamics. We will apply a combination of physics informed deep learning, and PDE solution methods to demonstrate improved generality over state of the art approaches.

4. **Apply these new techniques to the eVTOL transition maneuver problem.** This objective will be an application of the novel techniques developed and will serve to demonstrate their effectiveness in the context of an eVTOL transition maneuver. We will conduct trajectory optimization using the accelerated VPM simulations, and control synthesis using the learned Koopman operator.

# 4   Proposed Research

As outlined in the previous section, the proposed work will have four main thrusts. First, the VPM will be applied to an eVTOL transition optimization problem to demonstrate feasibility of the optimization approach and highlight the need for high fidelity modeling in this case. Next, the method of applying deep learning to accelerate VPM simulation through time step reduction will be developed and tested. After successful VPM accelerations are

achieved, the focus of the project will turn toward improving the generality of the Koopman learning problem, specifically for VPM dynamics. Finally, these innovations will be utilized to optimize trjectories and perform active feedback control for the eVTOL transition maneuver to demonstrate improved performance and reduced computational cost.

## 4.1   Trajectory optimization

For trajectory optimization, we represent the aircraft as a set of dynamic states $\mathbf{x}$, and control inputs $\mathbf{u}$ and we express the discrete time evolution of the aircraft states as

$$\mathbf{x}_{k+1} = F(\mathbf{x}_k, \mathbf{u}_k) \tag{1}$$

where $k$ denotes the current time step, and $F$ is the function that advances the state to time step $k+1$.

We then formulate the following optimization problem

$$
\begin{aligned}
\underset{\mathbf{x}_k, \mathbf{u}_k}{\text{minimize}} \quad & \sum_{k=0}^{N} \mathbf{u}_k^T R \mathbf{u}_k \\
\text{subject to} \quad & \mathbf{x}_{k+1} = F(\mathbf{x}_k, \mathbf{u}_k) \\
& \mathbf{x}_0 = \mathbf{x}_{\text{init}} \\
& \mathbf{x}_N = \mathbf{x}_{\text{des}}
\end{aligned}
\tag{2}
$$

The objective is a quadratic penalty on control effort that is determined by the weight matrix $R$. The system must adhere to the discrete dynamic equations $\mathbf{x}_{k+1} = F(\mathbf{x}_k, \mathbf{u}_k)$, and it must start at a predetermined state $\mathbf{x}_{\text{init}}$ and end at a desired state $\mathbf{x}_{\text{des}}$. Therefore solving this optimization problem will mean that the system successfully maneuvered to the desired location while expending the least ammount of control effort possible.

This problem is highly nonlinear due to the system dynamics $F$, which include unsteady wake mixing from the VPM. To make the problem tractable, we construct a local linear approximation of the dynamics using a first-order Taylor series expansion:

$$\mathbf{x}_{k+1} = F(\mathbf{x}_k, \mathbf{u}_k) \approx F(\mathbf{x}^*, \mathbf{u}^*) + A(\mathbf{x}_k - \mathbf{x}^*) + B(\mathbf{u}_k - \mathbf{u}^*) \tag{3}$$

where $A$ and $B$ are the Jacobians of the $F$ with respect to the states and inputs, evaluated about a nominal point $(\mathbf{x}^*, \mathbf{u}^*)$. For convenience, we rearrange and write

$$\mathbf{x}_{k+1} = A\mathbf{x}_k + B\mathbf{u}_k + C \tag{4}$$

where $C = F(\mathbf{x}^*, \mathbf{u}^*) - A\mathbf{x}^* - B\mathbf{u}^*$

This now represents a linear approximation of the nonlinear dynamics, but it is only valid close to $\mathbf{x}^*$ and $\mathbf{u}^*$. For an eVTOL transition, this is problematic because final desired state is cruise, which is very different from hover. To remedy this, we update the linear model at every time step, and allow $A$, $B$ and $C$ to vary with time. The equation now becomes

$$\mathbf{x}_{k+1} = A_k\mathbf{x}_k + B_k\mathbf{u}_k + C_k \tag{5}$$

And the optimization problem becomes

$$\underset{\mathbf{x}_k, \mathbf{u}_k}{\text{minimize}} \qquad \sum_{k=0}^{N} \mathbf{u}_k^T R \mathbf{u}_k$$

$$\text{subject to} \quad \mathbf{x}_{k+1} = A_k \mathbf{x}_k + B_k \mathbf{u}_k + C_k \tag{6}$$

$$\mathbf{x}_0 = \mathbf{x}_{\text{init}}$$

$$\mathbf{x}_N = \mathbf{x}_{\text{des}}$$

Because this formulation features a quadratic objective and linear constraints, it is classified as a quadratic program (QP) and can be solved analytically. However, the linear dynamics described by $A_k$, $B_k$, and $C_k$ are an approximation that is only valid close to the original trajectory that was used to construct the taylor series. If we use this model to solve the QP problem we will get an updated trajectory that is potentially different from the original one, and the model will no longer be valid. Therefore, iteration is required to converge to the true optimal solution. The procedure to iteratively solve the full nonlinear optimization problem is as follows:

1. Provide an initial guess trajectory.

2. Linearize the dynamics at each point along the trajectory to obtain $A_k$, $B_k$, and $C_k$.

3. Solve the QP problem defined in 6 to obtain a new trajectory.

4. Repeat steps 2–3 until convergence is achieved.

This procedure allows us to solve arbitrary nonlinear trajectory optimization problems, but solving it in the VPM presents additional challenges that need to be overcome. The main challenge associated with VPM-based trajectory optimization is derivative computation. The jacobian matrices $A_k$ and $B_k$ are computed as

$$A_k = \left. \frac{\partial F}{\partial \mathbf{x}} \right|_{\mathbf{x}_k^*, \mathbf{u}_k^*} \qquad B_k = \left. \frac{\partial F}{\partial \mathbf{u}} \right|_{\mathbf{x}_k^*, \mathbf{u}_k^*} \tag{7}$$

These derivatives are obtained via automatic differentiation (AD) by propagating derivatives through the VPM simulation, which is computationally expensive. The fine temporal resolution required in VPM often leads to simulations with thousands of time steps, further increasing the cost of derivative calculations along the trajectory.

To reduce this computational burden, $A_k$ and $B_k$ are not computed at every time step. Instead, they are evaluated at a limited number of key points along the trajectory, exploiting the observation that the Jacobians typically change gradually as the aircraft transitions from hover to cruise.

Once $A_k$ and $B_k$ have been calculated at these select points, the missing values at intermediate time steps are estimated by fitting a curve using least-squares regression. This approach substantially reduces the number of derivative computations required, thereby lowering overall computational cost.

Another challenge presented by the VPM is the fact that the linear model matrices $A_k$, $B_k$, and $C_k$ are generally very noisy. This noise is inherant in an unsteady aerodynamic model because vortical structures in the flow are shed from rotors that are spinning at a very high

RPM. The high blade passage frequency causes rapidly oscillating unsteady forces, which is completely consistent from a physics stand point, but nonetheless causes numerical issues for the optimization and makes it difficult to achieve convergence.
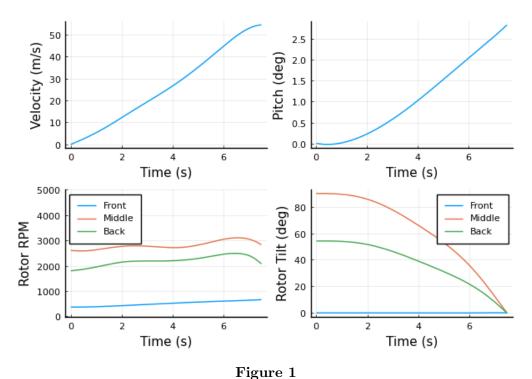
In order to improve the numerical performance of the optimization, we seek to filter out the high frequency content of the VPM. This is partly achieved through the regression of the derivative values described previously - fitting a relatively low order curve to the values $A_k$ $B_k$, and $C_k$ ensures a smooth progression in time. However, there is still a significant ammount of noise in these matrices across optimization iterations, which causes irregular optimizer steps, and hinders convergence.

To address this, we apply an exponential moving average of the matrices to smooth out the noise from iteration to iteration

$$\bar{A}_k = \alpha A_k + (1 - \alpha)\bar{A}_k \tag{8}$$

where $\bar{A}_k$ is the tracked average of the derivatives across iterations, $A_k$ is the newly computed derivative, and $\alpha$ is a tuning parameter which regulates the level of filtering.

These techniques have been successfully applied to the trajectory optimization of a tilt-rotor eVTOL inspired by the Joby S4 aircraft. The results of the trajectory optimization are presented in Fig. 1 which shows that the aircraft was able to reach a target velocity of 55 meters per second while maintaining constant altitude by gradually tilting its rotors forward.



**Figure 1**

As part of this study, we compare the force predictions from the VPM with those of a lower fidelity vortex lattice method (VLM) and show that the VLM does predict the correct trends, but does not capture the influence of a morphing wake during transition. This

discrepancy is shown in Fig. 2 where the two methods disagree by over 1000 N during early transition.
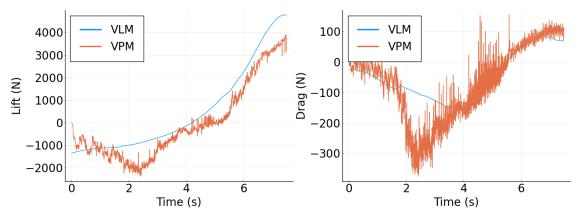


**Figure 2**

## 4.2   Neural time step reduction

The techniques outlined above present a feasible approach to trajectory optimization in the VPM by overcoming key challenges such as noise and the high cost of derivatives. However, the above optimization took roughly 428 hours to complete, which is a major limitation on the practicality of this approach. To address this issue, we will present a novel methodology that aims to accelerate the VPM by reducing the time steps necessary to solve the governing differential equations. While a major motivation of this method is directed toward accelerating the VPM, it is completely general and can be applied to any ordinary differential equation (ODE).

Consider a general system of ODEs

$$\frac{d\mathbf{x}}{dt} = f(\mathbf{x}) \tag{9}$$

If the dynamics of the system are fast, then small time steps will be necessary to resolve the correct trajectory, and if they are slow, then larger time steps may be sufficient, and we can simulate accurately with fewer of them.

The best metric for determining whether the dynamics are fast or slow is the spectral radius of the system's Jacobian, defined as

$$\rho\left(\frac{df}{d\mathbf{x}}\right) = ||\boldsymbol{\lambda}||_\infty \tag{10}$$

where $||\boldsymbol{\lambda}||_\infty$ is the magnitude of the largest eigenvalue of $\frac{df}{d\mathbf{x}}$

We desire a dynamic system which has a small spectral radius, and therefore slow dynamics, so that we can simulate the system accurately with only a small number of time steps. To achieve this, we will leverage two neural networks $\phi$, and $\psi$ that together form an autoencoder pair.

8

$$\mathbf{z} = \phi(\mathbf{x}) \quad \mathbf{x} = \psi(\mathbf{z}) \tag{11}$$

where $\mathbf{z} \in \mathbb{R}^m$ is a learned latent space, and is of higher dimension than $\mathbf{x} \in \mathbb{R}^n$: $m > n$. Then, by the chain rule,

$$\frac{d\mathbf{z}}{dt} = \frac{d\phi}{d\mathbf{x}} \frac{d\mathbf{x}}{dt} \tag{12}$$

Both $\frac{d\phi}{d\mathbf{x}}$ and $\frac{d\mathbf{x}}{dt}$ are functions of $\mathbf{x}$, but we can substitute $\mathbf{x} = \psi(\mathbf{z})$ to obtain

$$\frac{d\mathbf{z}}{dt} = \frac{d\phi}{d\mathbf{x}}(\psi(\mathbf{z})) \frac{d\mathbf{x}}{dt}(\psi(\mathbf{z})) \tag{13}$$

We have now expressed the dynamic system entirely in $\mathbf{z}$ space, and it is completely general because it was derived from the known dynamics of the system and the chain rule.

The next step is to train $\phi$ and $\psi$ to enforce slow dynamics in the latent representation. We define a loss function to be the spectral radius of the new dynamic system's Jacobian.

$$L_d = \rho\left(\frac{d\dot{\mathbf{z}}}{d\mathbf{z}}\right) \tag{14}$$

If we want the dynamics to be slow over a large domain in $\mathbf{z}$ space, then we need to penalize the spactral radius of the Jacobian at many points $\mathbf{z}_k$ in that domain

$$L_d = \frac{1}{N}\sum_{k=0}^{N}\rho\left(\frac{d\dot{\mathbf{z}}}{d\mathbf{z}}\Big|_{\mathbf{z}_k}\right) \tag{15}$$

The other loss function that we need to define is the standard autoencoder reconstruction loss to ensure that we learn a valid latent representation

$$L_r = \frac{1}{N}\sum_{k=0}^{N}||\mathbf{x}_k - \psi(\phi(\mathbf{x}_k))||_2^2 \tag{16}$$

The autoencoder pair is then trained to minimize the combined loss function

$$\underset{\phi,\psi}{\text{minimize}} \quad \frac{1}{N}\sum_{k=0}^{N}\rho\left(\frac{d\dot{\mathbf{z}}}{d\mathbf{z}}\Big|_{\mathbf{z}_k}\right) + ||\mathbf{x}_k - \psi(\phi(\mathbf{x}_k))||_2^2 \tag{17}$$

This method has been shown to be successful in reducing the number of time steps necessary to simulate the following ODE

$$\frac{dx}{dt} = -15x \tag{18}$$

which has eigenvalue $\lambda = -15$. Because the dynamics are relatively fast, a time step of 0.01 was necessary to accurately simulate the trajectory using euler integration. An autoencoder was trained using the above loss function, and then simulated accurately with a time step of 0.1 - an order of magnitude larger than what was required for the original system. As shown in Fig. ?? we simulated this new system and then reconstructed the original states using the decoder network. The new system was shown to perform much better than the original system when using a large time step of 0.1
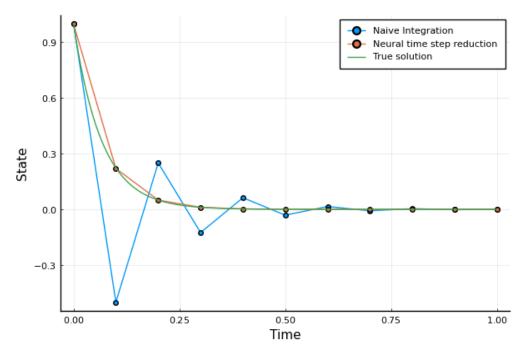
**Figure 3**

## 4.3   Koopman learning

The neural time step reduction method described above has the potential to significantly speeed up VPM simulations, and enable their use in iterative optimization loops. However, on its own, neural time step reduction is unlikely to achieve the speedups necessary for real time control, which requires many full simulations every second. In order to facilitate real time VPM-based control, we seek linear models that can be used to solve the optimal control problem analytically.

In order to achieve this, we leverage Koopman operator theory, which states that for any dynamic system

$$\frac{d\mathbf{x}}{dt} = f(\mathbf{x}) \tag{19}$$

there exists a mapping $\mathbf{z} = \phi(\mathbf{x})$ such that

$$\frac{d\mathbf{z}}{dt} = K\mathbf{z} \tag{20}$$

where $K$ is the Koopman operator and represents the globally linear dynamics of $\mathbf{z}$ space. Koopman operator theory states that such a map exists, but that it is potentially infinite dimensional.

To overcome this challenge, most current approaches apply machine learning to learn a finite dimensional approximation to the Koopman operator. For the mapping $\mathbf{z} = \phi(\mathbf{x})$ one can use existing basis functions like Hermite polynomials, but it's generally better to train a neural network to learn the best mapping possible.

10

Because current methods typically rely on trajectory data simulated or experimentally obtained beforehand, there is always a risk of overfitting to the specific trajectories in the data set. Because of this we are pursuing multiple options to potentially improve the generality of the Koopman learning process.

### 4.3.1 Method of characteristics

One way to approach learning the linear Koopman dynamic system is by considering the set of partial differential equations that arise from applying the chain rule to the learned latent space.

$$\frac{d\mathbf{z}}{dt} = \frac{d\mathbf{z}}{d\mathbf{x}}\frac{d\mathbf{x}}{dt} = \frac{d\mathbf{z}}{d\mathbf{x}}f(\mathbf{x}) \tag{21}$$

Setting this equal to 20 gives

$$\frac{d\mathbf{z}}{d\mathbf{x}}f(\mathbf{x}) = K\mathbf{z} \tag{22}$$

which is a linear partial differential equation that an be routinely solved using the method of characteristics. The method of characteristics solves for $z$ along characteristic curves by reducing the problem to a set of ordinary differential equations parameterized by a variable $t$

$$\frac{d\mathbf{x}}{dt} = f(\mathbf{x}), \ \ \frac{d\mathbf{z}}{dt} = K\mathbf{z} \tag{23}$$

This is the expected result and tells us that as $\mathbf{x}$ evolves according to $f(\mathbf{x})$, $\mathbf{z}$ evolves according to $K\mathbf{z}$. By itself, this is not useful yet because in order to solve for $\mathbf{z}$ we need to simulate the original dynamic system.

We can go further by solving the $\mathbf{z}$ ODE analytically as a function of $t$

$$\mathbf{z} = e^{Kt}\mathbf{z}_0 \tag{24}$$

where $\mathbf{z}_0$ is the initial condition. Ideally we would like to solve for $\mathbf{z}$ as a function of $\mathbf{x}$ rather than $t$, but due to the nonlinearity of the system, we cannot solve for $t(\mathbf{x})$ analytically. Instead, we propose approximating $t(\mathbf{x})$ using a neural network. This acts as a "time-to-reach" function, predicting how long it takes to reach a certain point in $\mathbf{x}$ space from a given initial point.

$t(\mathbf{x})$ alone only applies along a specific trajectory. In order to describe the full solution, we also need to identify which trajectory a point belongs to. To address this, we introduce another scalar valued network $l(\mathbf{x})$ to act as a "label" for each trajectory. Given a point in $\mathbf{x}$ space, $l(\mathbf{x})$ identifies which trajectory we are on, and $t(\mathbf{x})$ tells how far along that trajectory we are.

With these two functions defined, we have

$$\mathbf{z} = e^{Kt(\mathbf{x})}\mathbf{z}_0(l(\mathbf{x})) \tag{25}$$

where $\mathbf{z}_0$ is a function that accepts the trajectory label and returns the initial conditions in $\mathbf{z}$ space. This can be any function, but it should be injective to ensure unique initial conditions for each label value.

For training, we aim to find $t(\mathbf{x}$ and $l(\mathbf{x})$ satisfying

$$\begin{aligned} \nabla t^T f(\mathbf{x}) &= 1 \\ \nabla l^T f(\mathbf{x}) &= 0 \end{aligned} \tag{26}$$

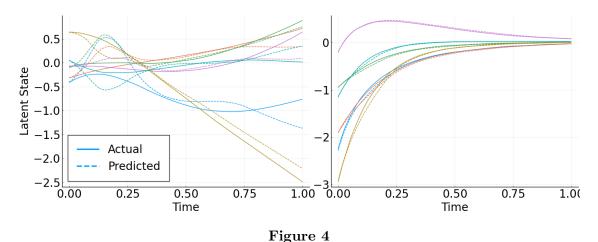In other words, $t$ increases unitarily along the trajectories, and $l$ is constant along the trajectories.

If initial conditions are known we can also apply boundary conditions. If $\mathbf{x}_0$ defines all the initial conditions, then we can enforce

$$\begin{aligned} t(\mathbf{x}_0) &= 0 \\ l(\mathbf{x}_0) &= \omega(\mathbf{x}_0) \end{aligned} \tag{27}$$

where $\omega(\mathbf{x}_0)$ is a known function defining the trajectory labels. For example, if $\mathbf{x} = (x, y)^T$ and we define initial conditions along the line $x = 0$, then the boundary conditions are $t(0, y) = 0$ and $l(0, y) = y$. This means that trajectories start on the $y$-axis, and are labeled by their $y$ value.

This method was used to learn a solution to the Koopman problem for a simulated glider undergoing a transient maneuver before stabilizing. The results where compared to the standard method of simply defining a vanilla neural network for $\mathbf{z}(\mathbf{x})$. In both methods, the latent states were simulated from initial conditions using the learned Koopman operator $K$, and compared to their "true" values which where encoded directly from the original trajectory. As shown in Fig. 4, the method of characteristics approach outperforms the standard method for the same number of collocation points.



Figure 4

### 4.3.2 Projection onto a neural basis

Leveraging the method of characteristics is potentially more genearlizable than the current state of the art in Koopman learning, but it still relies on obtaining trajectory data, and

may be subject to overfitting. Therefore we pursue a secondary approach directed toward constructing the Koopman operator without relying on data at all.

Given a set of basis functions $\phi_n(\mathbf{x})$ we can can represent a function $f(\mathbf{x})$ by an expansion in the basis

$$f(\mathbf{x}) = \sum_{n=0}^{\infty} a_n \phi_n(\mathbf{x}) \tag{28}$$

where $a_n$ is the projection of the function onto the $n$th basis over the domain of interest

$$a_n = \int_D f(\mathbf{x})\phi_n(\mathbf{x})d\mathbf{x} \tag{29}$$

The Koopman operator can be thought of a matrix of these projection values. If we have the discrete dynamic system $\mathbf{x}_{k+1} = F(\mathbf{x}_k)$, we want to find a matrix $K$ such that

$$\phi(\mathbf{x}_{k+1}) = \phi(F(\mathbf{x}_k)) = K\phi(\mathbf{x}_k) \tag{30}$$

So we need to construct the function $\phi(F(\mathbf{x}_k))$ by an expansion in the basis $\phi(\mathbf{x}_k)$. The Koopman operator $K$ is therefore a projection of the desired function onto this basis

$$K_{ij} = \int_D \phi_i(F(\mathbf{x}))\phi_j(\mathbf{x})d\mathbf{x} \tag{31}$$

This construction is desireable because it does not rely on data at all, and instead gives an analytic form of the best possible Koopman operator for a given set of bases.

There are two main challenges of this approach. The first is that the basis functions $\phi_n$ may not be the ideal basis for expressing the underlying dynamics, which could lead to large errors unless $\phi_n$ is infinite.

To address this issue, we will define $\phi_n$ as a neural network and train it to learn the best possible basis to work in for the specific problem. As a measure for how good the basis is, we examine the values of the projection coefficients, which reveal how much of the original function was captured by the projection. This is mathematically expressed through Parseval's theorem which states that for the expansion of a function $f(x)$ a complete basis,

$$||f||^2 = \sum_{n=0}^{\infty} |a_n|^2 \tag{32}$$

If this relation is true, then all of the "energy" of the function $f(x)$ is captured in the coefficients, and the function is exactly reconstructed. Applied to the Koopman problem, this means we can define a loss function

$$L_i = ||\phi_i(F(\mathbf{x}))||^2 - \sum_{j=0}^{N} |K_{ij}|^2 \tag{33}$$

Minimizing this error over all the rows of the Koopman operator will yield the best possible basis.

The second big challenge is that the integrand $\phi_j(\mathbf{x})\phi_i(F(\mathbf{x}))$ from 31 is potentially a very nonlinear and complex function that cannot be integrated analytically. We can integrate

numerically over the domain, but if the $\mathbf{x}$ is high dimensional, this quickly becomes extremely expensive.

We address this issue by recognizing that any complex function $f(x)$ can be described by a computational graph which consists of the composition of many simple operations $f(x) = f_n \circ \cdots \circ f_3 \circ f_2 \circ f_1 \circ x$.

Finding the projections for just $f_1$ is potentially easier if $f_1$ is a simple operation

$$\phi(f_1(x)) = K_1\phi(x) \tag{34}$$

Then, if we find the projections for $f_2$,

$$\phi(f_2(x)) = K_2\phi(x) \tag{35}$$

we can construct the projections for $f_2 \circ f_1$ because we have

$$\phi(f_2(f_1(x))) = K_2\phi(f_1(x)) = K_2K_1\phi(x) \tag{36}$$

We can therefore construct the full Koopman operator of $f$ by

$$\phi(f(x)) = K_n \cdots K_2K_1\phi(x) = K\phi(x) \tag{37}$$

This approach has the potential to be arbitrarily general because we relied on projection of our function onto a custom basis instead of training data. However, it is currently only in the very early stages of development and remains untested. It is also possibly susceptible to very large errors because each projection is an approximation, and the error accumulated over a large computational graph can be very significant unless it is trained sufficiently to minimize that error. For this reason, we will pursue both the projection approach and the method of characteristics approach to achieve the best outcome.

# 5 Anticipated Contributions

The proposed work, as outlined in the previous section, aims to bridge the gap between high fidelity simulation, and iterative optimization and control applications through several key contributions:

1. A practical demonstration of VPM-based trajectory optimization, including a comparative analysis that underscores the importance of unsteady modeling during eVTOL transition phases.

2. The development of a neural time-step reduction methodology applicable to any general set of ODEs, which will substantially accelerate high-fidelity simulations while preserving accuracy.

3. An enhancement of the generalizability of the Koopman learning process, enabling more robust linearization of nonlinear dynamics and supporting real-time control of complex, high-fidelity systems.

Together, these efforts will create a cohesive framework that unites high-fidelity aerodynamic modeling, machine learning, and control theory. By overcoming the computational and modeling barriers that have traditionally separated simulation from real-time decision making, this research will make it possible for optimization and control algorithms to directly leverage unsteady aerodynamic effects. In doing so, it lays the foundation for a new generation of aircraft design and control strategies—ones that are not only faster and more efficient, but also fundamentally more adaptive and aware of the physics that govern their motion.

# References

[1] S. Bakhshi, S. Kaneko, J. Martins, and A. Sharma, "Coupled trajectory and aerostructural optimization of a uav with modular morphing wings," 07 2025.

[2] S.-G. Lee, H.-H. Yang, R. Addo-Akoto, and J.-H. Han, "Transition flight trajectory optimization for a flapping-wing micro air vehicle with unsteady vortex-lattice method," *Aerospace*, vol. 9, p. 660, 10 2022.

[3] R. Tian, L. Li, W. Wang, X. Chang, S. Ravi, and G. Xie, "Cfd based parameter tuning for motion control of robotic fish," *Bioinspiration and Biomimetics*, vol. 15, 02 2020.

[4] N. Moradi, C. Wang, and F. Mafakheri, "Urban air mobility for last-mile transportation: A review," *Vehicles*, vol. 6, no. 3, pp. 1383–1414, 2024. [Online]. Available: https://www.mdpi.com/2624-8921/6/3/66

[5] S. Xiang, A. Xie, M. Ye, X. Yan, X. Han, H. Niu, Q. Li, and H. Huang, "Autonomous evtol: A summary of researches and challenges," *Green Energy and Intelligent Transportation*, vol. 3, no. 1, p. 100140, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2773153723000762

[6] W. Raza and R. S. Stansbury, "Noise prediction and mitigation for uas and evtol aircraft: A survey," *Drones*, vol. 9, no. 8, 2025. [Online]. Available: https://www.mdpi.com/2504-446X/9/8/577

[7] P. Shukla, S. Shukla, and A. Kumar Singh, "Trajectory-prediction techniques for unmanned aerial vehicles (uavs): A comprehensive survey," *IEEE Communications Surveys and Tutorials*, vol. 27, no. 3, pp. 1867–1910, 2025.

[8] S. S. Chauhan and J. R. R. A. Martins, "Tilt-wing evtol takeoff trajectory optimization," *Journal of Aircraft*, vol. 57, no. 1, pp. 93–112, 2020. [Online]. Available: https://doi.org/10.2514/1.C035476

[9] L. Panish and M. Bacic, "Tiltwing evtol transition trajectory optimization," *Journal of Aircraft*, vol. 62, no. 1, pp. 81–93, 2025. [Online]. Available: https://doi.org/10.2514/1.C037862

[10] J. Park, I. Kim, J. Suk, and S. Kim, "Trajectory optimization for takeoff and landing phase of uam considering energy and safety," *Aerospace Science and Technology*, vol. 140, p. 108489, 06 2023.

[11] M. Liu, Z. Su, J. Zhu, F. Guo, and Y. You, "Flight analysis and optimization design of vectored thrust evtol based on cooperative flight/propulsion control," *Aerospace Science and Technology*, vol. 149, p. 109143, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1270963824002761

[12] D. Sarojini, M. Ruh, A. J. Joshy, J. Yan, A. Ivanov, L. Scotzniovsky, A. Fletcher, N. Orndorff, M. Sperry, V. Gandarillas, I. Asher, J. Chambers, H. Gill, S. Lee, Z. Cheng,

G. Rodriguez, S. Zhao, C. Mi, T. Nascenzi, and J. Hwang, "Large-scale multidisciplinary design optimization of an evtol aircraft using comprehensive analysis," 01 2023.

[13] S. Kaneko and J. R. Martins, "Simultaneous optimization of design and takeoff trajectory for an evtol aircraft," *Aerospace Science and Technology*, vol. 155, p. 109617, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1270963824007466

[14] S. Kaneko and J. Martins, "Simultaneous design and trajectory optimization strategies for computationally expensive models," *AIAA Journal*, vol. 63, 01 2025.

[15] D. Thompson, "Robust control of an evtol through transition with a gain scheduling lqr controller," Master's Thesis, University of Maryland, College Park, Maryland, 2020.

[16] A. Gupta, K. Wolfe, G. Perrotta, and J. Moore, "Real-time planning and control with a vortex particle model for fixed-wing uavs in unsteady flows," 2025. [Online]. Available: https://arxiv.org/abs/2509.16079

[17] H. Yang and R. Morales, "Robust full-envelope flight control design for an evtol vehicle," *AIAA Scitech 2021 Forum*, 2021. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.2021-0254

[18] M. Korda and I. Mezić, "Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control," *Automatica*, vol. 93, pp. 149–160, 2018.

[19] I. Abraham, G. D. L. Torre, and T. D. Murphey, "Model-based control using koopman operators," *Robotics: Science and Systems*, 2017.

[20] V. Zinage and E. Bakolas, "Koopman operator based modeling for quadrotor control on se(3)," 2021, arXiv preprint.

[21] R. Anderson, J. Willis, J. Johnson, A. Ning, and R. W. Beard, *A Comparison of Aerodynamics Models for Optimizing the Takeoff and Transition of a Bi-wing Tailsitter*, 2021.

[22] A. C. Tagg, R. Anderson, C. Joseph, and A. Ning, "Trajectory optimization of evtol and conventional aircraft: A comparative analysis of vortex particle method and vortex lattice + blade element momentum theory," *AIAA AVIATION FORUM AND ASCEND 2024*. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.2024-3587

[23] W.-Q. Q. T.-J. Z. Y.-Q. C. Q. Wang, K.-F. He and K.-Y. Wu, "Unsteady aerodynamics modeling for flight dynamics application," *Acta Mechanica Sinica*.

[24] J. Dean, S. Morton, D. Mcdaniel, J. Clifton, and D. Bodkin, "Aircraft stability and control characteristics determined by system identification of cfd simulations," 08 2008.

[25] S. Görtz, D. Mcdaniel, and S. Morton, "Towards an efficient aircraft stability and control analysis capability using high-fidelity cfd," *Collection of Technical Papers - 45th AIAA Aerospace Sciences Meeting*, vol. 18, 01 2007.

[26] J. P. Thomas, E. H. Dowell, and K. C. Hall, "Modeling viscous transonic limit cycle oscillation behavior using a harmonic balance approach," *Journal of Aircraft*, vol. 41, no. 6, pp. 1266–1274, 2004. [Online]. Available: https://doi.org/10.2514/1.9839

[27] N. Bhoir and S. N. Singh, "Output feedback nonlinear control of an aeroelastic system with unsteady aerodynamics," *Aerospace Science and Technology*, vol. 8, no. 3, pp. 195–205, 2004. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1270963803001378

[28] G. Perrotta, L. Scheuer, Y. Kopel, M. Basescu, A. Polevoy, K. Wolfe, and J. Moore, "Planning and control for a dynamic morphing-wing uav using a vortex particle model," 2023. [Online]. Available: https://arxiv.org/abs/2307.02371

[29] M. Collotta, G. Pau, and R. Caponetto, "A real-time system based on a neural network model to control hexacopter trajectories," pp. 222–227, 2014.

[30] C. Sabater, P. Stuermer, and P. Bekemeyer, "Fast predictions of aircraft aerodynamics using deep learning techniques," 08 2021.

[31] Abucide, K. Portal-Porras, U. Fernandez-Gamiz, E. Zulueta, and A. Teso-Fz-Betoño, "A data augmentation-based technique for deep learning applied to cfd simulations," *Mathematics*, vol. 9, p. 1843, 08 2021.

[32] W. Hwang, D. Lee, J. Shin, K. Cho, and S. Choi, "Aerodynamic flow analysis using conditional convolutional autoencoder in various flow conditions and application to cfd-based design optimization," *Advances in Aerodynamics*, vol. 7, 09 2025.

[33] M. Raissi, P. Perdikaris, and G. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0021999118307125

[34] J. Lin, S. Chen, H. Yang, Q.-f. Jiang, and J. Liu, "A physics-informed neural network-based aerodynamic parameter identification method for aircraft," *Physics of Fluids*, vol. 37, 02 2025.

[35] V. Sharma, A. Rauch, and V. Raman, "Accelerating cfd simulations with super-resolution feedback-informed adaptive mesh refinement," 01 2025.

[36] H. Gao, L. Sun, and J.-X. Wang, "Super-resolution and denoising of fluid flow using physics-informed convolutional neural networks without high-resolution labels," *Physics of Fluids*, vol. 33, no. 7, p. 073603, Jul. 2021. [Online]. Available: http://dx.doi.org/10.1063/5.0054312

[37] G. Jarry, R. Dalmau, X. Olive, and P. Very, "A neural ode approach to aircraft flight dynamics modelling," 2025. [Online]. Available: https://arxiv.org/abs/2509.23307

[38] J. Ma, Y. Li, J. Tu, Y. Zhang, J. Ai, and Y. Dong, "Development and implementation of physics-informed neural ode for dynamics modeling of a fixed-wing aircraft under icing/fault," *Guidance, Navigation and Control*, vol. 04, no. 01, p. 2350024, 2024. [Online]. Available: https://doi.org/10.1142/S2737480723500243