Modeling and Control of an Ornithopter for Diving

Cameron J. Rose, Parsa Mahmoudieh, and Ronald S. Fearing

***Abstract*— This paper details a method for identifying a set of piece-wise affine linear models that can be used for control design for flapping-winged flight. The paper focuses on diving maneuvers as the application for these models. The flight conditions during the dive are segmented into separate dynamically similar regions, and least-squares is used to estimate affine linear models for each modeling region. These models are used to compute the reachability sets that satisfy recovery conditions for safe diving. The point within the dive to begin recovery was determined by checking the current pose for inclusion in the backward reachable set. Using this control method, 2.2 meter dives were achieved at a success rate of 60 percent. The data-driven automatic modeling techniques and controller design processes can be extended to additional flight maneuvers, provided sufficient previous data have been collected for model generation of those maneuvers.**

I. Introduction

Bio-inspired flapping-winged robotics is a rapidly growing field that is of interest to roboticists as well as biologists. The aerodynamics of flapping-winged flight are nonlinear and complex and are difficult to model. The flapping of the wings creates an unsteady airflow around the control surfaces of the flier, increasing the complexity of the aerodynamics associated with the surfaces [1][2][3][4]. An understanding of the behavior of these fliers in free flight is necessary for successful control. Modeling the complete aerodynamics and dynamics of flapping-winged fliers is computationally intensive, so performing predictive control online is not possible on an autonomous millirobotic platform due to computational limitations. Therefore, a controller algorithm that can be updated at a high sampling rate is necessary.

One method of understanding the flight behavior of these fliers involves modeling the wing motion during each wing stroke, e.g. using blade element theory [5][6][7]. Another method involves multi-body modeling to account for the changing mass distribution as the flier flaps its wings [8][9][10].

To accomplish on-board model-based control in 10 gram scale fliers, a simplified representation of the aerodynamics is necessary. With flight models that are more mathematically tractable, fast control schemes can be designed to achieve maneuvers that have not previously been studied.

Models computed using learning approaches can often have less online computational overhead than strictly dynamic models, which are nonlinear for flapping-winged flight. The trade-off is that model learning approaches can require large training data sets and high offline computational overhead for high-dimensional models. Nguyen-Tuong et. al. conducted a survey of some of the current model learning approaches for robot control [11]. For diving maneuvers, it is necessary to have multi-step modeling capabilities to prevent crashes, which can require a level of accuracy that necessitates larger data sets and increased robustness.

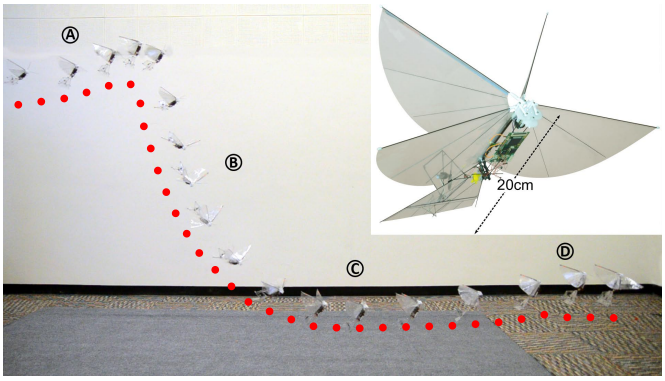

Fig. 1: Sample H^{2}Bird dive maneuver path and a close up of the H^{2}Bird (inset). Letters A-D indicate the initiation of the dive, the beginning of the recovery, the minimum height in the dive, and the final height regulation, respectively.

For this aerodynamic modeling, it is necessary to collect data that can be used to estimate the behavior of the MAV over time. Common approaches to develop these models include averaging the flight behavior over the wing beat period, and using linear low-dimensional models to predict the flight behavior over time. It has been shown that using time averaged aerodynamic data is a valid approximation over a given wing stroke [12][13]. Wind tunnels are often used to measure aerodynamic properties of robotic fliers, although some of the degrees of freedom are constrained by the mounting mechanism. A simulation of insect flight for the Robofly project uses aerodynamic models based upon wind tunnel measurements [14][15]. In addition, mounted sensor measurements have also been used to measure the aerodynamic properties of wings for modeling by Khan and Agrawal [16]. Lee and Han recently implemented a non-contact magnetic suspension and balance system to control the attitude of an experimental model [17]. This setup allows for data collection and controller verification using selected degrees of freedom.

An alternative method to using stationary wind tunnel data for modeling is collecting free flight telemetry data using a motion tracking system. Grauer et al. utilized this method to

*This material is based upon work supported by the National Science Foundation under Grant No. 1427096.

The authors are with the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720 USA {c_rose, ronf}@eecs.berkeley.edu, parsa.m@berkeley.edu

create a dynamic model for their "Slow Hawk" ornithopter [18]. The flight data collected was used to fit parameters to a multi-body dynamic model, and wind tunnel tests were used to determine the associated aerodynamic model. This method, however, involves the fitting of many model parameters. Faruque et al. also used system identification to generate linear models of the pitch and roll dynamics of their insect-inspired micro-flier [19]. Their models were of a lower complexity than Grauer et al., and they used the models to inform their on-board stabilizing avionics package.

The flapping-winged behaviors that have been previously studied are typically limited to height regulation, vision-based path following, and flight stabilization. While these behaviors have been extensively investigated, the methods described in this paper are used as an approach to develop models and controllers for non-equilibrium maneuvers.

One approach that is often used is Proportional-Integral-Differential (PID) control for height regulation or path following. Baek et al. used on-board orientation estimation along with a camera to seek and fly towards an LED using their flapping-winged MAV, the iBird [20][21]. They predicted the position of the LED using a Kalman filter and controlled the yaw and height using a PID controller.

Ma et al. used mixed-model based control methods for height regulation of a robotic fly, developed by Wood [22][23]. They used a PD controller to regulate the attitude of the flier, and calculated the angular reference to the attitude controller using the lateral position reference error. The altitude controller was designed using the linearized dynamics of the flier about hover. The controllers were decoupled to reduce the constraints on the sensitive attitude and lateral position controllers.

A vision-based approach was utilized by the researchers at the Delft University of Technology with their flapping flier, the Delfly II [24]. de Croon et al. used a camera mounted on the front of the Delfly to compute optical flow and texton histograms for texture detection [25]. Both methods were used to estimate the time to collision with an obstacle. A human controlled the height, and open loop yaw inputs in the vision loop were used for obstacle avoidance. In this case, the vision inputs determined the necessary control, rather than a specific model or controller.

Another approach was used by Moore and Tedrake to control the perching behavior of a fixed wing aircraft [26]. The researchers used LQR Trees, a control method that Tedrake developed [27]. They designed a feedback controller using trajectory optimization and local linear feedback. The set of controllers over the local trajectory regions are computed within regions of attraction branching from the space of initial conditions. The regions of attraction are computed using time-varying Lyapunov functions.

Gerdes et al. studied aerobatic flight using independently controlled wings on their Robo-Raven flapping-winged MAV [28]. The authors used programmable servos for independent wing actuation. They investigated aerobatic maneuvers for their flier such as tight turns, backflips, and dives by pre-programming the wing actuator motions based upon simulations.

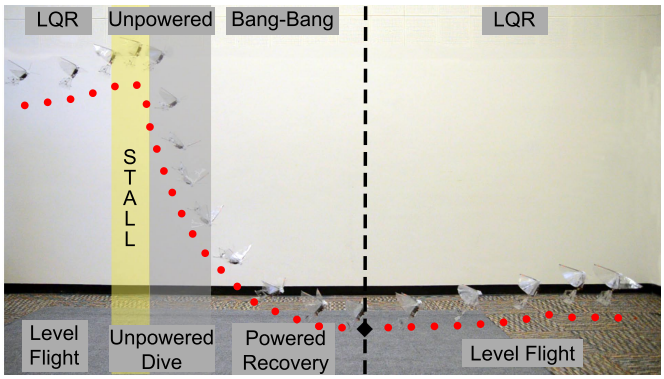

Fig. 2: Diagram of the control sections of the entire diving maneuver. The black diamond marks the lowest point in the dive.

II. Piece-wise Affine Modeling

The diving behaviors of the H^{2}Bird were modeled using a piece-wise affine discrete time model with a 0.025 second time step. The state data were segmented into diving and recovery sections, and each section was segmented into several linear models using K-means and least squares regression, a similar approach used by Buchan et al. [29]. Using these linear models, reachability analysis determined when to switch control behaviors within the maneuver to reach the goal height.

A. Data Collection

Diving data were collected in free flight using a Vicon motion capture system to collect translational position, velocity, and pitch angle data from the H^{2}Bird during the dive. There were 29 dives in total, and each experiment consisted of:

1) Release the H^{2}Bird by hand from one side of the tracking space.
2) Wait for the initial release transient flight to stabilize.
3) Manually begin the unpowered dive portion of the maneuver.
4) Manually begin the powered recovery portion of the maneuver.

During the unpowered dive portion of the maneuver, the H^{2}Bird stops flapping, and the wings remain pressed together. The elevator is commanded to a neutral position of 20 degrees and the yaw controller remains active. Since the maneuver is within the sagittal plane, the yaw motion for these experiments was not modeled.

During the powered recovery portion of the maneuver, the wings flap at a maximum frequency of approximately 20 Hz, and the elevator is set to its maximum deflection of 60 degrees. The 160 mAh LiPo battery was replaced after every 3 trials to eliminate the effects of battery drain on the experiments.

An illustration depicting the different sections of the diving maneuver is in Figure 2. Both the gray and yellow

shading represent the unpowered dive portion of the maneuver. The yellow shading alone represents an initial stall period immediately after the wings are turned off to start the dive. During this region, the forward velocity rapidly decreases and the nose pitches up, causing a rapid increase in downward velocity before the nose pitches back down. This stall period before a glide slope is reached is aerodynamically complex. Everything after the gray shading and before the vertical dashed line is considered the recovery part of the maneuver. The lowest point of the maneuver, which is the final dive height, is indicated by the black diamond. Hereafter, all references to the beginning and final heights of the dive will represent the height at the beginning of the unpowered portion and the lowest point during the recovery portion, respectively. The means and standard deviations of the terminal conditions of the maneuver for the 29 trials are in Table I.

Variable	Mean	Standard Deviation	Min	Max
X Velocity [m/s]	3.53	0.41	2.55	4.56
Z Velocity [m/s]	0.03	0.10	-0.14	0.22
Pitch [rad]	0.88	0.07	0.76	1.07

TABLE I: Table of the mean, minimum, maximum, and standard deviation of the terminal conditions for relevant variables for the 29 model training dives.

Ground effect was not modeled during the diving trials, although there were dives that came very close to the ground. As a result, ground effect could be aiding the recovery of the H^{2}Bird from some of the dives, as ground effect increases lift and decreases drag on an airfoil as it comes into close proximity with a fixed surface. This phenomenon could have some effect on the end result of the modeling used to design the controllers.

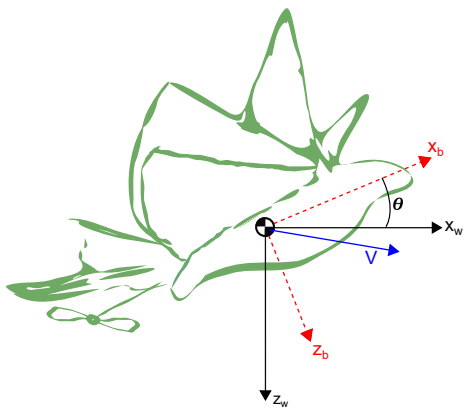

Fig. 3: Free body diagram of the H^{2}Bird for the relevant state variables for the discrete-time models [20].

B. Data Segmentation

The flight trajectory data were segmented into several regions and linearized about the regions to reduce the numerical complexity of the aerodynamics. The state of the H^{2}Bird is represented as

$$\mathbf{X} = \begin{bmatrix} x_w & z_w & \dot{x}_w & \dot{z}_w & \theta \end{bmatrix} \quad (1)$$

In the vector, x_w is the forward position in world coordinates, z_w is the vertical position in world coordinates, θ is the pitch angle, $\dot{x}_w$ is the horizontal velocity in world coordinates, and $\dot{z}_w$ is the vertical velocity in world coordinates. Figure 3 is a diagram of the relevant state variables and their associated directions.

The model chosen for a particular data segmentation region should not be dependent upon the translational position of the robot, so the segmentation routine was used on only a portion of the state vector:

$$\mathbf{X}_p = \begin{bmatrix} \dot{x}_w & \dot{z}_w & \theta \end{bmatrix} \quad (2)$$

K-means clustering was used to segment the data, which partitions the data into k clusters, each used as a separate region for linearization [30]. An initial set of k region centers $m_1^1, \ldots, m_k^1$ from all of the observed state data in all of the diving trials is randomly selected to begin the segmentation process. In the assignment step, each data point x_p is then assigned to the closest region center, according to Equation 3, where S_i^t is the ith region cluster at time t.

$$S_i^t = \{x_p \ni \|x_p - m_i^t\|^2 \leq \|x_p - m_j^t\|^2 \forall j, 1 \leq j \leq k\} \quad (3)$$

After the assignment step, the region centers are recalculated, according to Equation 4, as the mean of the regions of clustered data points.

$$m_i^{t+1} = \frac{1}{|S_i^t|} \sum_{x_j \in S_i^t} x_j \quad (4)$$

The update and assignment steps are run iteratively 100 times. To account for bad seeding of the initial randomly selected region centers, the entire k-means algorithm is run 10 times and the regions that minimize the sum of the squared distances from each data point to its associated region cluster are selected:

$$\arg\min_S \sum_{i=1}^{k} \sum_{x \in S_i} \|\mathbf{x} - \mu_i\|^2 \quad (5)$$

The results of the k-means segmentation of the two portions of the dive are graphically represented in Figure 4, with the model regions for the dive on the left and the regions for the recovery on the right. Each color represents a different region of the state space described in Equation 2.

C. Piece-wise Affine Model Generation

The least squares algorithm was used to fit linear models to each region generated by the k-means segmentation. The piece-wise linear discrete-time models fit to the data in each region are of the form:

$$x_{t+1} = A_i x_t + B_i u_t + f_i \quad (6)$$

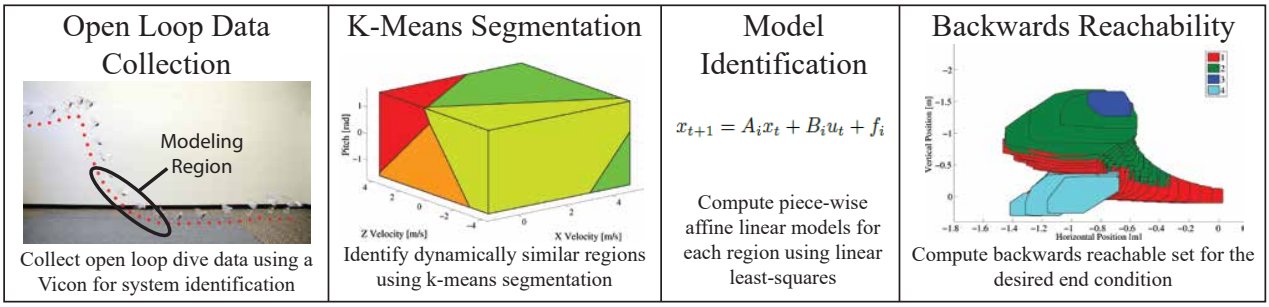

Fig. 4: Overview of the data collection, segmentation, model generation, and reachable set computation processes. The result of this process is used to determine when to switch from the unpowered dive to the recovery control mode.

where $x_{t+1} \epsilon \mathbb{R}^{5x1}$ is the state vector described by Equation 1 at the next time step, $x_t \epsilon \mathbb{R}^{5x1}$ is the state vector at the current time step, $u_t \epsilon \mathbb{R}^{2x1}$ is the input vector, $A_i \epsilon \mathbb{R}^{5x5}$ is the state evolution matrix, $B_i \epsilon \mathbb{R}^{5x2}$ is the input matrix, and $f_i \epsilon \mathbb{R}^{5x1}$ is an affine portion of the state evolution equation, for a modeling region i. The inputs in the vector u_t are the wing flap frequency and the elevator angle. A time step of 0.025 seconds was used for the modeling.

Since the first two variables of the state vector are the x and z position, the state update equations for that portion of the state space are:

$$\begin{aligned} x_{t+1} &= x_t + T_s * \dot{x}_t \\ z_{t+1} &= z_t + T_s * \dot{z}_t \end{aligned} \tag{7}$$

where T_s is the time step. Therefore, the model is only being fit to the reduced state vector in Equation 2, the same state vector used for the k-means segmentation. This ensures that the segmentation and models are independent of the position.

The model parameters A_i, B_i, and f_i for each model region i are the solutions to the following least squares optimization problem:

$$\mathbf{X}_{t+1} = \begin{bmatrix} A & B & f \end{bmatrix}_i \begin{bmatrix} \mathbf{X} \\ \mathbf{u} \\ \mathbf{1} \end{bmatrix}_t = \beta_i \bar{\mathbf{X}}_t \tag{8}$$

β_i is the unknown model parameter matrix for all of the points in region S_i that satisfy Equation 3. $\mathbf{X}_{t+1} \epsilon \mathbb{R}^{3xN}$ is the vector of all N next states in the region i, collected into the observation matrix of the least squares problem. $\mathbf{X} \epsilon \mathbb{R}^{3xN}$ is the vector of all N current states in region i, $\mathbf{u} \epsilon \mathbb{R}^{2xN}$ is the vector of all N inputs in region i, and $\mathbf{1} \epsilon \mathbb{R}^{1xN}$ is a vector of N ones, collected into the data matrix $\bar{\mathbf{X}}_t$ of the least squares problem. The least squares problem was solved for each region i in each the dive and recovery regions for one- to ten-region k-means segmentations.

The Mahalanobis distance metric was used to determine the appropriate number of regions to model the motion of the H^{2}Bird:

$$\sqrt{\frac{\sum_{i=1}^{N} \left(\beta_i \bar{\mathbf{X}}_t - \mathbf{X}_{t+1}\right)^\top * \Sigma_x * \left(\beta_i \bar{\mathbf{X}}_t - \mathbf{X}_{t+1}\right)}{N}} \tag{9}$$

N is the number of data points in the particular model region i and Σ_x is the co-variance matrix of the state measurements. The Mahalanobis distance was used as the comparison metric because it relates vector quantities in different spaces based on the distribution of values in a signal. 80 percent of the measured data was used as the training set for the models, and the other 20 percent was used as the validation set. The distance values for the validation sets are in Table II. The distance metric for the "Null" model is in the first column of the table for comparison with the other fitted models. The "Null" model is defined as:

$$\mathbf{X}_{t+1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{X}_t \tag{10}$$

For ease of computation, the models were computed using the number of regions that provided an appreciable decrease in error compared to a fewer number of regions.

# of Regions	Null	1	2	3
Dive	0.113	0.0309	0.0188	0.0208
Escape	0.501	0.113	0.0859	0.0586
	4	5	6	7
	0.0205	0.0215	0.0212	0.0209
	0.0540	0.0516	0.0470	0.0440

TABLE II: Table of the Mahalanobis distance for 1 through 7 regions for the dive and escape portions of the maneuver.

III. Reachability Analysis

To determine when to switch from the uncontrolled dive to the bang-bang recovery controller, the backward reachable sets were computed over a fixed horizon from the goal state. The set of goal states were defined as the region of observed terminal conditions at the lowest vertical point in the dive. This point is indicated by a black diamond in Figure 2. To conduct the reachability analysis, the set of possible goal states were defined as the minimum and maximum of the observed terminal velocities and pitch. The goal position was defined as zero for the controller and thereby the goal terminal region was defined as 0.1 meters above and below zero for the horizontal and vertical position. An equality constraint at zero for the goal position constrained

the problem too much and did not provide the needed result for the analysis.

The backwards reachable set is defined as:

$$B\left(x_f, U, t\right) = \{x_0 \,\epsilon\, X \;:\; \exists u \,\epsilon\, U, \exists t \,\epsilon\, [0,t] \;\; s.t.\; x(t) = x_f\} \tag{11}$$

The backwards reachable set is the set of all states x_0 such that there exists an input $u \,\epsilon\, U$ that can drive the collection of states x_0 into the collection of states x_f in t time. The backwards reachable set for the goal region was computed for 16 time steps at a time step of 0.025 seconds, for a total of 0.4 seconds. The total collection of all of the polytopes for the reachable sets for each step in time is the total set of states that can reach the target height in $t \leq 0.4s$ time. This set boundary was used as a guard condition for switching control schemes from the unpowered dive portion to the recovery portion of the maneuver. The total reachable set within 0.4 seconds for the goal set is shown in Figure 5. When the trajectory of the H^{2}Bird during the dive enters this set, the control mode switches into the recovery mode.

IV. Online Control

A. Control Implementation

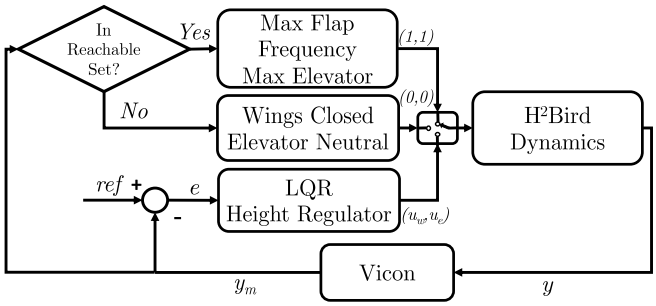

Fig. 6: Controller block diagram of the diving maneuver.

The controllers for the diving maneuver are sectioned into two distinct pieces: the uncontrolled dive and the powered recovery. The location within the dive of each section of the control is graphically depicted in Figure 2 by the labels on the top of the diagram. To start the maneuver, the H^{2}Bird begins the unpowered dive and recovers from the dive at a time dictated by the models described in Section II. A block diagram of the control implementation is in Figure 6. The controller switches from an initial height regulation controller when the dive is initiated by the user. The transition from the unpowered dive to the recovery controller occurs when the estimated state, y_m, of the H^{2}Bird is within the reachability polytope described in Section III. Each controller section along with the transition conditions are described in the following subsections.

1) Uncontrolled Dive: The dive portion of the maneuver is uncontrolled; the wings stop flapping and the elevator is commanded to a neutral position of 20 degrees upwards deflection. The position at which the dive begins is marked as the starting position of the maneuver.

2) Recovery Control: The set generated by the reachability analysis is used as the hybrid guard condition to determine the transition point between the uncontrolled dive and recovery portions of the maneuver. The reachability set for the desired goal set is numerically represented by a collection of N polytopes of the form $A_i x_t \leq b_i$, where x_t is the current state, $A_i \,\epsilon\, \mathbb{R}^{M\text{x}5}$, and $b_i \,\epsilon\, \mathbb{R}^{M\text{x}1}$. M depends on the number of faces in the polytopes and varies polytope to polytope. If the current state is within the reachability set, the control scheme is switched to the recovery controller. This check can be formally stated as:

$$\text{if } \exists \mathcal{P}_i \mid \mathcal{P}_i = A_i x \leq b_i,\; A_i x_t - bi \leq 0 \Rightarrow \text{recover} \tag{12}$$

$\mathcal{P}_i$ is the ith polytope of the reachable set.

The recovery controller is a bang-bang controller that stays active until the vertical velocity becomes positive. During the recovery, the wings are flapped at maximum frequency and the elevator is at its full upward deflection. The moment the vertical velocity becomes positive is recorded as the final dive height, and the controller switches back to the height regulator.

V. Experiments and Discussion

Experiments on the dive recovery control scheme were conducted using a Vicon motion capture system to collect flight telemetry. The Vicon was also used in the control loop as the vertical and horizontal distance and velocity measurement [1].

A. Recovery Control

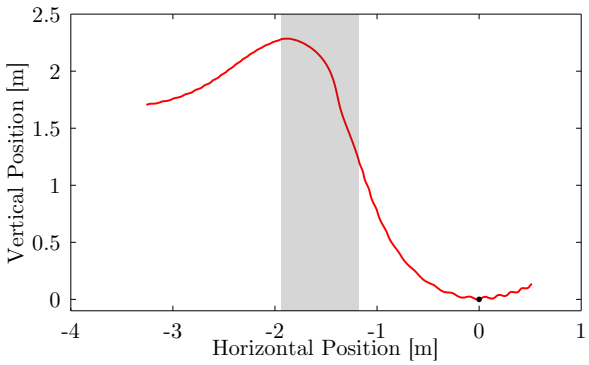

Fig. 7: Position data for a single trial of the H^{2}Bird diving experiment. The gray shading represents the unpowered portion of the dive. The black marker indicates the conditions at the lowest vertical position in the dive.

Experiments using various desired dive heights were conducted to determine the effectiveness of the recovery controller. Each trial consisted of releasing the H^{2}Bird by

[1]Unfortunately, the H^{2}Bird does not have any sensors to measure position and velocity onboard.

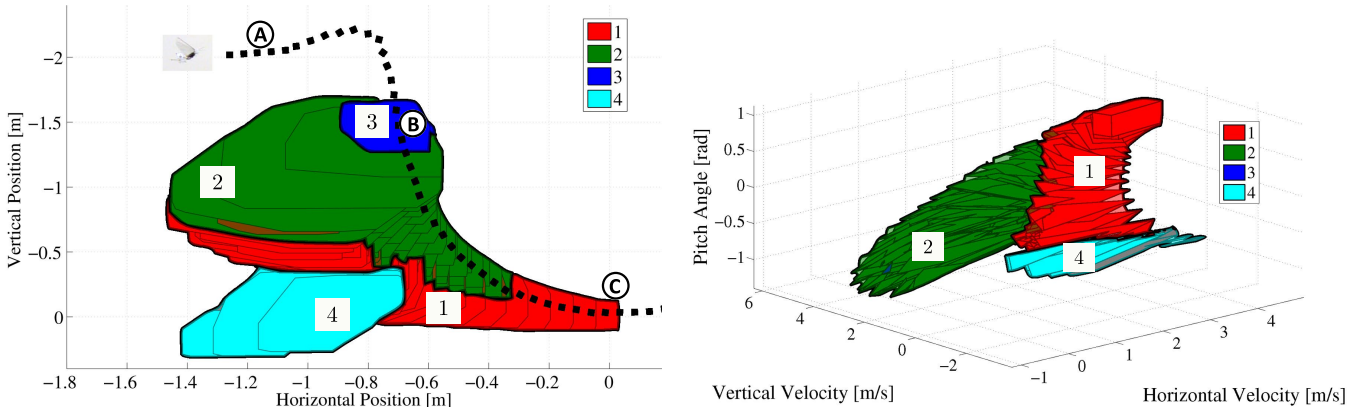

Fig. 5: The backwards reachable polytopes 16 time steps, or 0.4 seconds, from the terminal set in the horizontal and vertical position space on the left and the horizontal velocity, vertical velocity, and pitch angle space on the right. Letters A - C in the left image correspond to the positions in the trajectory marked by letters A - C in Figure 1.

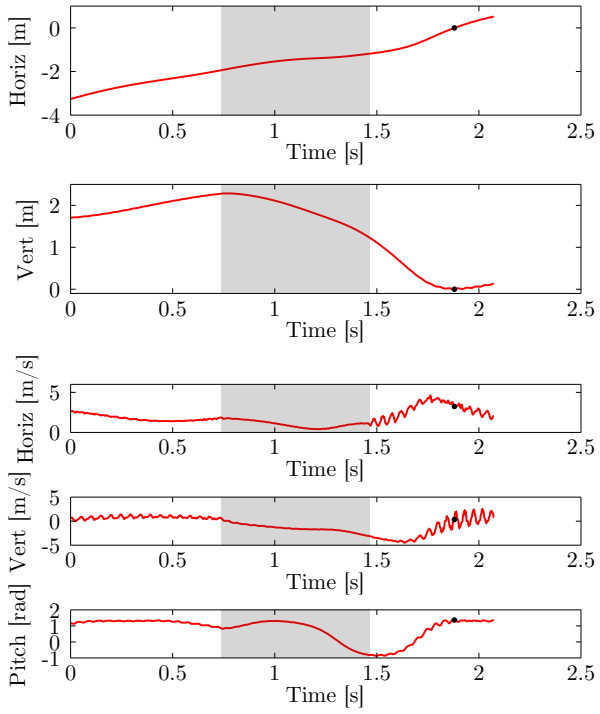

Fig. 8: Telemetry data for a single trial of the H^{2}Bird diving experiment. From the top are the horizontal position, vertical position, horizontal velocity, vertical velocity, and pitch angle. The gray shading represents the unpowered portion of the dive. The black marker indicates the conditions at the lowest vertical position in the dive.

hand, then triggering a dive to a particular set point. The rest of the maneuver is autonomous, and the existing yaw controller remained active to prevent lateral movement. A successful dive was classified as one that did not contact the ground, and a failed dive was classified as one that did. There were no requirements or constraints on post-dive behavior, although the H^{2}Bird ascended at the end of each trial. Target dive distances of 1.0, 1.5, and 2.0 meters were used. For each trial, the H^{2}Bird was released from approximately 2.6 meters above the ground with a 30 degree pitch angle set point, and the dive was not initiated until the initial launch transient ended. During each trial, the motor inputs from the H^{2}Bird and translational position, translational velocity, and pitch angle from the Vicon were recorded.

Desired Dist [m]	Succeed	Fail	Mean Dist [m]	Std Dev
1.0	5	4	2.2	0.09
1.5	8	5	2.2	0.1
2.0	0	2	N/A	N/A

TABLE III: Table of the number of successful trials, failed trials, mean dive distance, and standard deviations for each tested dive distance.

The data for a typical trial are presented in Figures 7 and 8. The gray shaded region indicates the unpowered dive portion of the experiment, and the recovery trigger point is at the rightmost edge of the gray region. The black marker indicates the lowest vertical point in the trajectory. In the parts of the trajectory where the wings are flapping shows oscillations in pitch and velocity due to wing artifacts because the velocity is a time derivative of the tracked position. The Vicon only provides angular position in axis-angle format and the translational position of the tracked body by default. Since the Vicon data was streaming directly to the control program, the built-in post-processing provided by the tracking software could not be used.

There were 24 total dive trials conducted: 9 at a 1.0 meter desired dive depth, 13 at a 1.5 meter depth, and 2 at a 2.0 meter depth. The results of the dives are summarized in Table III. For a 1.0 meter target dive distance, there was a success rate of approximately 55 percent, for a 1.5 meter target distance there was a 61 percent success rate, and for 2.0 meters there were no successful dives. The robot showed no signs of success at the 2.0 meter set-point and would hit the ground at a high rate of speed, eventually resulting in damage, so no more than two trials were conducted.

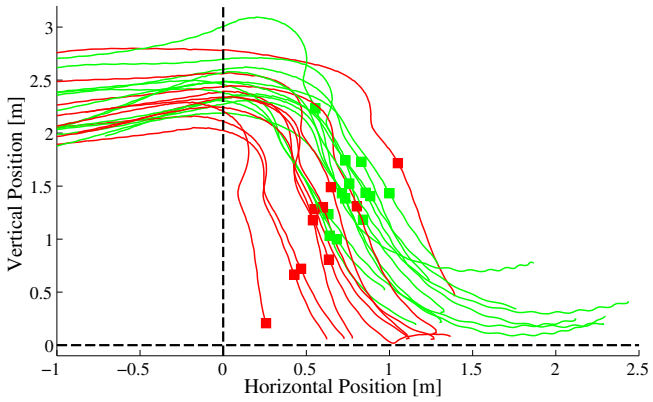

Fig. 9: The tracked paths for all of the dive experiments with successful trials in green and failed trials in red. The black vertical line is the starting point of the unpowered dive section and the horizontal line is the lower tracking limit of the Vicon. The squares represent the start of the recovery controller.

The total dive distance from the point that the dive is initiated (the wings stop flapping) to the lowest vertical point in the maneuver is also in Table III. Unfortunately, the Vicon tracking for some of the trials was of low quality in the final parts of 1 dive at 1.0 meters and 1 dive at 1.5 meters, so these trials were omitted. Although the H^{2}Bird managed to enter the backward reachable set in each of these trials and successfully recover, the mean dive distance was 2.2 meters for both 1.0 and 1.5 meter set-point. The reason for this was that the mean dive distance for the initial trials used to train the models was 1.8 meters with a standard deviation of 0.2 meters. The lack of diversity in the dive distances that the model was based upon resulted in a model that consistently yielded dive distances around 2.2 meters, regardless of whether the set-point was 1.0 or 1.5 meters. This problem could probably be remedied with a larger training set with increased diversity in observed dive heights.

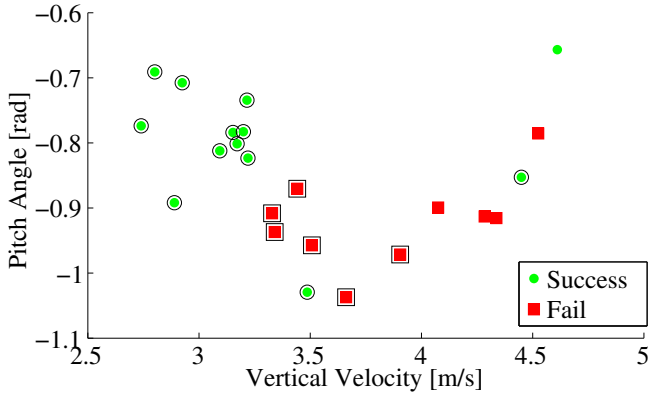

Fig. 10: The vertical velocity vs. the pitch angle at the time of recovery for successful (green circle) and failed (red square) trials.

The flight paths for all of the diving trials are in Figure 9.

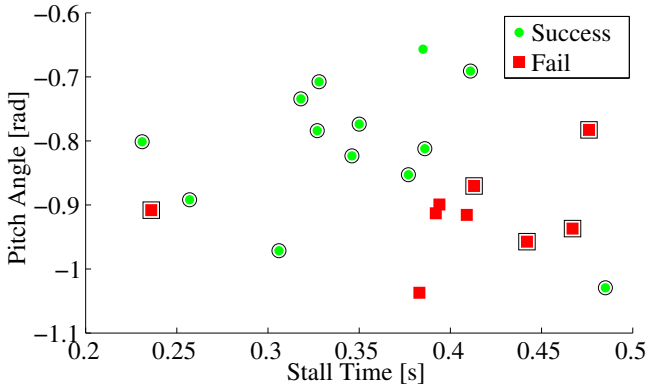

Fig. 11: The pitch angle at the recovery point and the time to a downward pitch angle during the unpowered dive for successful (green circle) and failed (red square) trials.

The vertical black line indicates the beginning of the unpowered portion of the dive, and the horizontal black line indicates the lower tracking limit of the Vicon, which is approximately the floor. The lowest recovery initiation height of successful dive was 1.0 meters. The failures that initiated recovery below this point can be attributed to attempts to recover too close to the floor.

The vertical velocity and the pitch angle at the recovery point for successful (green circle) and unsuccessful (red square) trials are plotted in Figure 10 to attempt to understand why some of the trials in which recovery was initiated above 1.0 meters failed. The markers with their respective shapes surrounded in black are the trials when recovery initiated above 1.0 meters. All of the failed trials exhibit both a high vertical velocity and high downward pitch angle at the time of recovery. When the wings begin flapping they will induce both a downward pitch moment and increase in velocity, which further exacerbates the problem. When this happens, the H^{2}Bird is likely to lose enough height, before reaching a positive pitch angle, to crash into the ground.

These recovery conditions can be caused by the properties of the stalling behavior shown in Figure 11. The x axis represents the amount of time spent in a stall in the unpowered dive, and the y axis represents the pitch angle at the recovery point. The stall time is the time elapsed between the maximum pitch angle during the stall and the first occurrence of a negative pitch angle. On average, the failed trials exhibit higher stall times, which lead to increased vertical velocities at the point of recovery. The stall region at the beginning of the dive is represented by the flight path diagram in Figure 2.

VI. Conclusions

To design computationally tractable models for online control during a diving maneuver, linear models were fit around clusters of flight conditions from a data set of sample manually controlled dives. The data sets were segmented into clusters of similar flight poses using k-means, and the models were fit using linear least squares. The models were then used to compute the backward reachable sets from a

goal set of previously experienced dive end poses. The point within the dive to begin recovery was determined using the backward reachable set by checking the current pose and simulated poses 0.1 seconds ahead for inclusion in the reachable set. Using this control method, 2.2 meter dives were achieved at a success rate of 60 percent. The majority of the failures can be attributed to extended stall conditions that result in dangerous vertical velocities and pitch angles during recovery. Another possible reason for the failures could be attributed to the chosen observations to compute the models. It is possible that observing the kinetic and potential energy in the system during the dives could provide some insight into good recovery conditions.

Linear piece-wise affine modeling of segments of flight conditions within a maneuver has proven to be an effective method for determining transition points between hybrid controllers. This method could be extended to incorporate models for chunks of additional maneuvers. These sections of flight could form a library of maneuver segments that could be stitched together to form flight patterns and trajectories that were not previously experienced by the robot. The controllers for each segment could be stored on-board to enable autonomous navigation or obstacle avoidance by picking maneuvers applicable to an observed situation. Since the models are linear, the computational overhead for on-board look-ahead simulation or computation of feedback controllers is reduced relative to complex nonlinear models.

ACKNOWLEDGMENT

The authors thank the members of the Biomimetic Millisystems Laboratory and the EECS community at the University of California, Berkeley for their advice and support.

REFERENCES

[1] H. Liu and K. Kawachi, "Leading-edge vortices of flapping and rotary wings at low Reynolds number." *Progress in Astronautics and Aeronautics*, vol. 195, pp. 275–285, 2001.

[2] R. Ames, O. Wong, and N. Komerath, "On the flowfield and forces generated by a flapping rectangular wing at low Reynolds number." *Progress in Astronautics and Aeronautics*, vol. 195, pp. 287–305, 2001.

[3] D. Lentink, G. F. van Heijst, F. T. Muijres, and J. L. van Leeuwen, "Vortex interactions with flapping wings and fins can be unpredictable," *Biology Letters*, vol. 6, pp. 394–397, 2010.

[4] G. C. H. E. de Croon, K. M. E. de Clercq, R. Ruijsink, B. Remes, and C. de Wagter, "Design, aerodynamics, and vision-based control of the DelFly," *International Journal of Micro Air Vehicles*, vol. Vol. 1, no. 2, pp. 71–97, June 2009.

[5] B. Cheng and X. Deng, "Translational and rotational damping of flapping flight and its dynamics and stability at hovering," *IEEE Transactions on Robotics*, vol. Vol. 27, no. 5, pp. 849–864, 2011.

[6] J. Han, J. Lee, and D. Kim, "Ornithopter modeling for flight simulation," in *Intl. Conf. on Control, Automation and Systems*, 2008, pp. 1773–1777.

[7] A. T. Pfeiffer, J. Lee, J. Han, and H. Baier, "Ornithopter flight simulation based on flexible multi-body dynamics," *Journal of Bionic Engineering*, vol. Vol. 7, no. 1, pp. 102 – 111, 2010.

[8] M. A. Bolender, "Rigid multi-body equations-of-motion for flapping wing MAVs using Kanes equations," in *AIAA Guidance, Navigation, and Control Conference*, 2009.

[9] J. Grauer and J. Hubbard, "Multibody model of an ornithopter," in *Journal of Guidance, Control, and Dynamics*, vol. 32, no. 5, Sept. - Oct. 2009, pp. 1675–1679.

[10] C. Orlowski and A. Girard, "Modeling and simulation of nonlinear dynamics of flapping wing micro air vehicles," *AIAA Journal*, vol. 49, no. 5, pp. 969–981, May 2011.

[11] D. Nguyen-Tuong and J. Peters, "Model learning for robot control: a survey," *Cognitive Processing*, pp. 1–22, 2011.

[12] Z. Khan and S. Agrawal, "Control of longitudinal flight dynamics of a flapping-wing micro air vehicle using time-averaged model and differential flatness based controller," in *American Control Conference*, 2007, pp. 5284–5289.

[13] L. Schenato, D. Campolo, and S. Sastry, "Controllability issues in flapping flight for biomimetic micro aerial vehicles (MAVs)," in *42nd IEEE Conf. on Decision and Control*, vol. 6, 2003, pp. 6441–6447.

[14] W. B. Dickson, A. D. Straw, C. Poelma, and M. H. Dickinson, "An integrative model of insect flight control," in *44th American Institute of Aeronautics and Astronautics Sciences Meeting*, 2006.

[15] S. Sane and M. Dickinson, "The aerodynamic effects of wing rotation and a revised quasi-steady model of flapping flight," *Journal of Experimental Biology*, vol. Vol. 205, no. 8, pp. 1087–1096, 2002.

[16] Z. A. Khan and S. K. Agrawal, "Force and moment characterization of flapping wings for micro air vehicle application," in *American Control Conference*, 2005.

[17] D. Lee and J. Han, "Flight controller design of a flapping-wing MAV in a magnetically levitated environmentl," in *Intl. Conf. on Robotics and Automation*, 2013.

[18] J. Grauer, E. Ulrich, J. Hubbard Jr., D. Pines, and J. Humbert, "System identification of an ornithopter aerodynamics model," in *AIAA Atmospheric Flight Mechanics Conference*, Aug. 2010.

[19] I. Faruque, P. Samuel, and J. Humbert, "Moment generation of stabilizing axes for insect-inspired flapping wing flight," *AIAA Atmospheric Flight Mechanics Conference*, June 2014.

[20] S. Baek and R. Fearing, "Flight forces and altitude regulation of 12 gram i-bird," in *IEEE Intl. Conf. on Biomedical Robotics and Biomechatronics*, 2010, pp. 454–460.

[21] S. Baek, F. G. Bermudez, and R. Fearing, "Flight control for target seeking by 13 gram ornithopter." in *IEEE Intl. Conf. on Intelligent Robots and Systems*, 2011, pp. 2674–2681.

[22] R. Wood, "The first takeoff of a biologically inspired at-scale robotic insect," *IEEE Trans. Robot*, 2008.

[23] K. Ma, P. Chirarattananon, S. Fuller, and R. Wood, "Controlled flight of a biologically inspired, insect-scale robot," *Science*, vol. 340, no. 6132, pp. 603–607, 2013. [Online]. Available: http://www.sciencemag.org/content/340/6132/603.abstract

[24] K. De Clercq, R. de Kat, B. Remes, B. W. van Oudheusden, and H. Bijl, "Aerodynamic experiments on DelFly II: Unsteady lift enhancement," *International Journal of Micro Air Vehicles*, vol. 1, no. 4, pp. 255–262, 2009. [Online]. Available: http://mav.sagepub.com/content/1/4/255.abstract

[25] G. de Croon, K. de Clercq, Ruijsink, Remes, and C. de Wagter, "Design, aerodynamics, and vision-based control of the DelFly," *International Journal of Micro Air Vehicles*, vol. 1, no. 2, pp. 71–97, 2009. [Online]. Available: http://dx.doi.org/10.1260/175682909789498288

[26] J. Moore and R. Tedrake, "Control synthesis and verification for a perching UAV using LQR-Trees," in *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, Dec 2012, pp. 3707–3714.

[27] R. Tedrake, "LQR-Trees: Feedback motion planning on sparse randomized trees," in *Papers of the fifth annual Robotics: Science and Systems conference*, July 2009.

[28] J. Gerdes, A. Holness, A. Perez-Rosado, L. Roberts, A. Greisinger, E. Barnett, J. Kempny, D. Lingam, C.-H. Yeh, H. A. Bruck, and S. K. Gupta, "Robo raven: A flapping-wing air vehicle with highly compliant and independently controlled wings," *Soft Robotics*, vol. 1, no. 4, pp. 275–288, December 2014.

[29] A. D. Buchan, D. W. Haldane, and R. S. Fearing, "Automatic identification of dynamic piecewise affine models for a running robot," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, Nov 2013, pp. 5600–5607.

[30] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*. Berkeley, Calif.: University of California Press, 1967, pp. 281–297. [Online]. Available: http://projecteuclid.org/euclid.bsmsp/1200512992