Bartholomew, J. P. A., Calway, A. D., & Mayol-Cuevas, W. (2015). Improving MAV control by predicting aerodynamic effects of obstacles. In 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2015) : Proceedings of a meeting held 28 September - 2 October 2015, Hamburg, Germany. (pp. 4826-4833). Institute of Electrical and Electronics Engineers (IEEE). DOI: 10.1109/IROS.2015.7354055

Improving MAV Control by Predicting Aerodynamic Effects of Obstacles

John Bartholomew, Andrew Calway and Walterio Mayol-Cuevas
Department of Computer Science, University of Bristol, UK

***Abstract*— Building on our previous work [1], in this paper we demonstrate how it is possible to improve flight control of a MAV that experiences aerodynamic disturbances caused by objects on its path. Predictions based on low resolution depth images taken at a distance are incorporated into the flight control loop on the throttle channel as this is adjusted to target undisrupted level flight. We demonstrate that a statistically significant improvement ($p << 0.001$) is possible for some common obstacles such as boxes and steps, compared to using conventional feedback-only control. Our approach and results are encouraging toward more autonomous MAV exploration strategies.**

I. Introduction

For many of their intended uses, Micro Air Vehicles (MAVs) must work indoors and in tight urban environments, operating in close proximity to obstacles. These obstacles will necessarily affect the airflow around the MAV, and so affect its flight, even without direct contact. A well known example of this is 'ground effect', which causes an increase in lift for wings or rotors that are close enough to a flat horizontal surface. For full scale aircraft, this effect is typically only relevant during take-off and landing, with most of the flight occurring far from any obstacle, but at the scale of MAVs, flying indoors or in 'urban canyons', obstacles are typically much closer (in proportion to the size of the vehicle) and are also more varied in their shape. This leads to more complex airflow effects, which are not so well known or studied.

Being able to predict the effect on the MAV of airflow around obstacles is potentially very useful. It could be used to improve flight control, or for higher level path planning decisions (*e.g.*, avoiding areas which are predicted to exhibit high turbulence). In this paper, we show how using predicted aerodynamic effects on varied shaped obstacles, observed from a distance, improves flight control accuracy. We concentrate on the disturbances caused by the MAV's rotors themselves interacting with scene structure, but the concept of utilizing learning for prediction from depth images is generalizable to include other external disturbances.

There has been a significant amount of work on control systems for quadrotor platforms (and other designs), concentrating on improving modelling and control of the MAV, independent of its flight environment. This has resulted in some very impressive capabilities in terms of accurately following paths and performing aggressive manoeuvres in free space (*e.g.*, [6, 5]). However, even the most sophisticated control systems are limited by the accuracy of their model of the platform's behaviour. Any system behaviour that is not

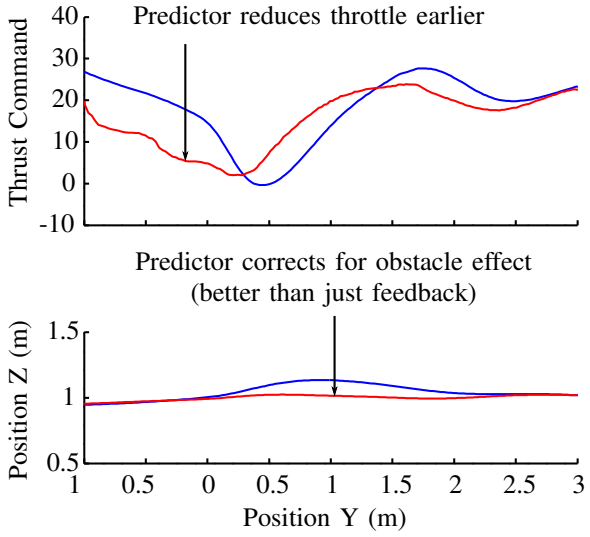

Fig. 1. Throttle command and measured flight path while passing over a box obstacle. Blue plot shows the behaviour of the system when using only normal feedback control (no prediction). Red plot shows the system when using feedback augmented with an extra term from the predictor.

captured by the control model will result in a 'disturbance' that must be corrected by feedback. This includes, for example, gusts of wind, interaction from humans (someone physically perturbing the vehicle's flight, as is commonly done during testing and tuning of the control system), and also the effects of aerodynamic interaction between the MAV and nearby obstacles. While a properly tuned control system should have no difficulty correcting for such unmodelled disturbances, that correction must necessarily happen *after* the MAV has deviated from the demanded flight path. This is true no matter how sophisticated the control system is: it is limited by the accuracy of its system model.

In order to reject disturbances without deviating from the demanded flight path, the control system must have knowledge of disturbances before they happen, *i.e.*, it must be able to predict them. One approach to making such predictions would be to perform real-time computational airflow simulation for the region around the MAV. While this is technically possible, it is currently impractical for real-time control, since airflow simulation is computationally very expensive. On the other hand, even if it was possible to run real-time airflow simulation onboard a MAV and in real-time, not being able to learn from previous experience

is wasteful. Another approach would be to try to create some simplified analytical model for the aerodynamic effect of obstacles, but such a model is likely to be quite inflexible in the types of obstacle for which it can make accurate predictions. A third approach, the approach taken in this paper, is to make predictions based on experience from prior flights over similar obstacles and use this information to make predictions useful for adjusting the controller.

This leads to the question of what the input to the predictor should be. A good candidate is to base the prediction on some descriptor of the shape of the MAV's environment. Shape is suitable since the airflow around an object primarily depends on its shape. For the prediction to be useful, it must be made at a distance, but there are several sensors that can provide such information: stereoscopic vision, a depth camera, a laser range scanner, and so on. In this work we use a structured light depth camera, but the prediction method we describe should be equally applicable to data from other sensors.

In section II we discuss previous work that has examined aerodynamic interactions between autonomous flying vehicles and their environment. Section III describes the equipment that we use and the controller on which we base our experiments. Section IV summarizes the prediction method we use, which was previously introduced in [1]. We proceed in section V to explain how we incorporate the predictor into the MAV's control system, and in section VI we discuss and analyse the effectiveness of our predictive controller. Finally, we conclude in section VII and note some areas where future investigation would be desirable.

II. Related Work

A MAV that is expected to perform exploration and navigation on unknown environments will face a series of challenges related to obstacle avoidance and dealing with disturbances both external and those resulting from its own influence on nearby obstacles.

While there is some work on using knowledge of potential wind disturbances on aerial vehicles, we note that this in general has received little attention. Perhaps as flying close to objects is not necessarily a strategy for vehicles other than small indoor UAVs.

Up-drafts, as caused by thermal convection in relative proximity to the ground, are commonly exploited by both birds and glider pilots to extend flight duration, and are starting to receive attention for their use in UAVs [7].

Cowling and colleagues in [3] explore the use of computational flow dynamics for simulated wind effects on building structures to perform path planning for a $>$1Kg plane. In such case, a simplified 2D analysis of a small urban area is used to enable rough on-board prediction of up-drafts to support efficient path planning flying at an altitude of 120m above ground. Beyond path planning, awareness of wind disturbance also has impact on flight formation and efficient trajectory following [2].

In the case of small UAVs or MAVs, and in particular those powered by rotors, both ground effect and ceiling effect can result in significant disturbances [10]. Ground effect is a well known effect that is usually modelled considering rotor diameter and a simplified scene of a flat surface [4].

Fig. 2. Some of the obstacles used in the experiments. First row: the arena (also shows the RGB-D sensor on tripod), 'box', 'steps'. Second row: 'ramp', 'table', 'table (cluttered)'. The quadrotor platform is also shown for relative scale. Obstacles are contained roughly within a $120 \times 80 \times 72 cm^3$ volume (WxDxH).

Using these simplifications, some works have started to incorporate awareness of such effects into the controller of MAVs. Nonaka *et al.* developed a ground-effect-aware altitude controller for a small coaxial configuration R/C helicopter [9]. They empirically propose a second order polynomial function model of ground effect conceding the difficulty of obtaining an analytic formulation. They perform characterisation and experiments of this effect in the neighbourhood of 0mm to 400mm from a flat surface. Similar work is considered in simulation to include external wind disturbance in [11].

Overall we note that i) aerodynamic effects are important for a MAV to be aware of, that ii) there is currently limited work in this area and that iii) critically, approaches to date concentrate on dealing with the effects at the control layer and that involving prediction as part of the control loop has not been explored in MAVs. We also note the extended use of the flat surface approximation for scene structure in these models, which contrasts with a small vehicle closely exploring or traversing a rich 3D environment.

In this paper, we make a first attempt at using predicted aerodynamic effects on a representative variety of 3D shapes perceived from a distance, in order to improve flight control accuracy for a MAV.

III. Experimental Setup

A. Equipment

We perform experiments in a flight space which is instrumented with a Vicon motion capture system. This system is used to track the MAV's location and orientation, both to provide the feedback signal for the control loop in the baseline cases, and for subsequent analysis of the MAV's behaviour when using our approach.

Our MAV is a generic off-the-shelf quadrotor kit, the DJI Flamewheels F450 with a DJI NAZA-M autopilot module and with $10''$, (Mod 1045) propellers. This receives a control signal from a laptop via a 2.4 GHz R/C transmitter. The control signal sent to the MAV has four channels: throttle,

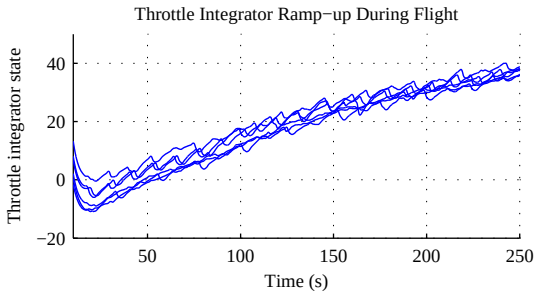

Fig. 3. PID integrator state for the throttle channel in a series of test flights. The general increasing trend is caused by the integrator compensating for the drop in battery voltage over the course of the flight. The higher frequency perturbations correspond with the individual passes over the obstacle. Note that even in the no-obstacle case, throttle changes are needed to maintain altitude while changing pitch to follow the demanded trajectory.

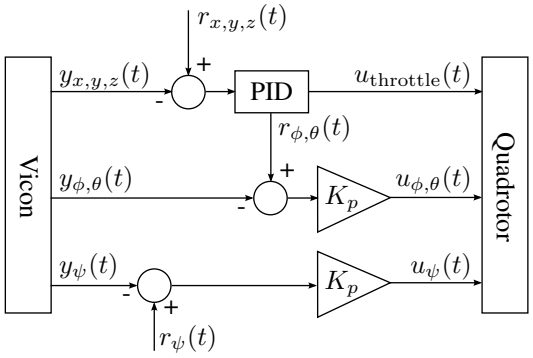

Fig. 4. Control system block diagram, using state signal y, control signal u, and reference (set-point) signal r. Signal components are position (x, y, z) and roll (ϕ), pitch (θ) and yaw (ψ). Some details (*e.g.*, wrap-around of orientation error) are omitted.

roll, pitch and yaw. The units for these channels are not well specified, particularly for the throttle channel (as explained below), but the roll, pitch and yaw channels control the demanded rate-of-rotation on those axes.

The throttle signal sent to the MAV effectively specifies a percentage of maximum thrust. Since the maximum thrust drops over the course of the flight as the voltage of the flight battery drops, the rotor speed (and therefore the actual thrust force exerted by the rotors) does not have a constant relation to the signal sent on the MAV's throttle channel. That is, over the course of a flight, the throttle signal must be increased to compensate for the drop in battery voltage. This is handled in our system by the integral term of the Proportional Integral Derivative (PID) controller used to control the MAV (this controller is described in Section III-B). The effect is illustrated in figure 3, which plots the integrator state over time for a set of example flights.

To capture geometric data about the MAV's environment, we use an Asus Xtion Pro RGB-D sensor (in these experiments, only the depth images are used). Since the payload capacity of our MAV is limited, we do not mount this sensor on the MAV itself, but instead use it to separately capture a static point-cloud for each obstacle.

The point-cloud for an obstacle is made from multiple depth images captured and registered with the Vicon coordinate system to form a single point-cloud for each test obstacle. This point-cloud can then be used during flight to generate synthetic depth views in real-time from the exact MAV's point of view. Note that this is a practical compromise and does not imply that only the original trajectories or that the sensor-object approaches used for mapping the 3D scene are the only ones used to train our system or in the live experiments. Instead, we are able to synthesize any 3D view for any trajectory and approach performed in real-time by the MAV.

B. Control System

The flight control system is split into three stages: rate control, attitude control and position control. The inner-most stage (rate control with auto throttle controller disabled) is run on-board the quadrotor itself: rate gyroscopes are used to stabilise the platform's rate of roll, pitch and yaw using proportional feedback. The outer two control stages are run on a laptop (off-board the platform), and are implemented as a Simulink model which communicates over the network with the Vicon motion capture system (to receive position and attitude feedback) and the R/C transmitter (to send the control signal to the quadrotor). This outer loop is run at 50 Hz. A block diagram for the off-board control system is shown in figure 4.

Attitude control is maintained using proportional feedback only, since the quadrotor responds quickly to attitude rate changes. The three attitude control axes (roll, pitch and yaw) are each treated as equivalent Single-Input Single-Output (SISO) systems. As a safety precaution, the demanded roll and pitch angles are limited to be in the range $[-30, 30]$ degrees.

Given a reference signal $r(t)$ and the platform attitude $y(t)$ (measured by the motion capture system), the attitude controller is a simple proportional controller which defines the roll, pitch and yaw rate control signals $u(t)$.

$$\begin{aligned} \text{wrap}(x) &= ((x+\pi) \mod 2\pi) - \pi \\ u_\phi(t) &= 400\ \text{wrap}(r_\phi(t) - y_\phi(t)) \\ u_\theta(t) &= 400\ \text{wrap}(r_\theta(t) - y_\theta(t)) \\ u_\psi(t) &= 400\ \text{wrap}(r_\psi(t) - y_\psi(t)) \end{aligned} \tag{1}$$

Positional control is maintained using a PID controller for each axis. A first-order low pass filter with a cut-off frequency of 3 Hz is applied to the derivative term to prevent measurement noise from affecting the control signal. For safety and to mitigate against integral wind-up, a saturation limit is placed on the integrator state.

The general form of the PID controller used is:

$$\begin{aligned} e(t) &= r(t) - y(t) \\ \text{PID}_{K_p,T_i,T_d}(c;t) &= \\ & K_p\left(e_c(t) + \frac{1}{T_i}\int_0^t e_c(\tau)d\tau + T_d\frac{d}{dt}e_c(t)\right) \end{aligned} \tag{2}$$

And the position controller for the quadrotor uses three such PID controllers:

$$\begin{aligned} u_z(t) &= \mathrm{PID}_{80,6,0.8125}(z;t) \\ r_\phi(t) &= \mathrm{PID}_{10,6,0.75}(x;t) \\ r_\theta(t) &= \mathrm{PID}_{10,6,0.75}(y;t) \end{aligned} \tag{3}$$

The gains used in these control equations were obtained experimentally in a similar way to the Ziegler-Nichols tuning method.

For each experiment, the quadrotor is commanded to follow a linear path over the test obstacle. This is repeated several times depending on the state of the flight battery. Prior to each pass over the obstacle, the position set-point is held constant while the vehicle's state is stabilised. This allows the integrators in the control system to find the current neutral point, which compensates for the gradual drop in battery voltage. The pass over the obstacle is only initiated once the vehicle's dynamic state (position and velocity) have been held within set error bounds (± 0.1 cm horizontal position error, ± 7.5 cm vertical position error, speed under 4 cm/s) for two seconds. Those error bounds were selected as giving adequate consistency in experimental conditions without spending too much time waiting for the platform's state to stabilise. Note that the error bounds cannot be reduced to zero, since there are always general air movements in the flight space that will perturb the platform's position.

In some of the experiments we are measuring open-loop behaviour of the vehicle in the vertical/throttle axis. For those experiments, the position hold is performed as normal, but during the linear pass over the obstacle, the throttle signal is held at the 'neutral' value as estimated by the integrator of the throttle PID controller (the integrator itself is also disabled during the pass).

IV. Prediction Method

Our prediction method follows [1], but we provide a brief summary here. We use a non-parametric regression method to predict the effect of obstacles based on training flights. The input to the regression is a depth image from the vehicle's point of view, and the output is the acceleration on the vehicle produced by its aerodynamic interaction with the obstacle. In this work we only predict the vertical component of the acceleration, but the method can be trivially extended to predict acceleration in three dimensions, or to incorporate angular impulses or other measurements of the obstacle's effect on the vehicle.

For clarity, it is worth explicitly noting that this method works by learning the correlation between the shape of an obstacle and its aerodynamic effect as the vehicle passes over it. The depth image acts as a descriptor of the obstacle shape and provides the basis for a metric to be applied to measure the similarity between different obstacle shapes. The acceleration measured during training or predicted for a new depth image must be the acceleration produced on the vehicle while it is in the region of space described by the depth image. This means the prediction is produced for some distance ahead of the vehicle's current location. Here, we use a prediction distance of 1.5 m.

The non-parametric regression method that we use is the Nadaraya-Watson estimator [8, 12] (also known as kernel regression). The form of this estimator is given in equation (4). Given a database of n training examples, each of which is a pair of a depth image $\mathbf{D}_i$ and a measured acceleration $\mathbf{y}_i$, we can estimate the acceleration $\tilde{\mathbf{y}}$ for a new image $\mathbf{D}'$.

$$w_i = \exp(-\alpha(\|\mathbf{D}' - \mathbf{D}_i\|_1)^2) \tag{4a}$$

$$\tilde{\mathbf{y}} = \frac{\sum_{i=1}^{n} \mathbf{y}_i w_i}{\sum_{i=1}^{n} w_i} \tag{4b}$$

The estimate is made by taking a weighted average of the training examples, computing the weights by applying a Gaussian kernel to the $\mathbf{L}^1$ distance between the new depth image and the depth image from the training example. This estimator has one tuning parameter α, which controls the bandwidth of the Gaussian kernel—the effect of this tuning parameter is discussed in [1].

A. Data Capture

As noted in section III-A, our depth images are synthesised during flight from a point-cloud which is captured separately. We synthesise depth images at a resolution of 40×60 pixels, with a field of view of $35°$ horizontal and $46°$ vertical. The vertical field of view here corresponds to that of the ASUS RGB-D sensor; though it would be easy to synthesise a view with a larger vertical field of view, using the field of view of the sensor makes it easier for future work to switch to a platform that carries this sensor rather than using synthesised depth views. A limited horizontal field of view is used so that the imaged region matches more closely with the region of space in front of the vehicle which will have the greatest aerodynamic effect (*i.e.*, the region of space for which we want to make a prediction).

Apart from field of view, another important factor of the depth images that we use is that they are 'stabilised' in orientation. That is, the virtual 'camera' used to synthesise the depth images takes its position from the position of the quadrotor (as measured by the motion capture system), but roll and pitch is ignored. This stabilisation is required because the image distance metric being used is not invariant to changes in feature location. If depth images were captured online rather than synthesised, then this stabilisation could be achieved quite easily using a gimbal mount for the RGB-D sensor (such stabilised gimbal mounts are commonly used on MAVs for photography), or by applying a rectifying homography transform to the captured depth images based on inertial measurement of the MAV's attitude.

To build the training database, we command the vehicle to fly over a set of example obstacles. These flights are run with the control system configured to hold the throttle signal constant during the pass over the obstacle. This allows us to measure the effect of the obstacle in terms of vertical acceleration, with some independence from the properties of the control system. Multiple trials are performed over each

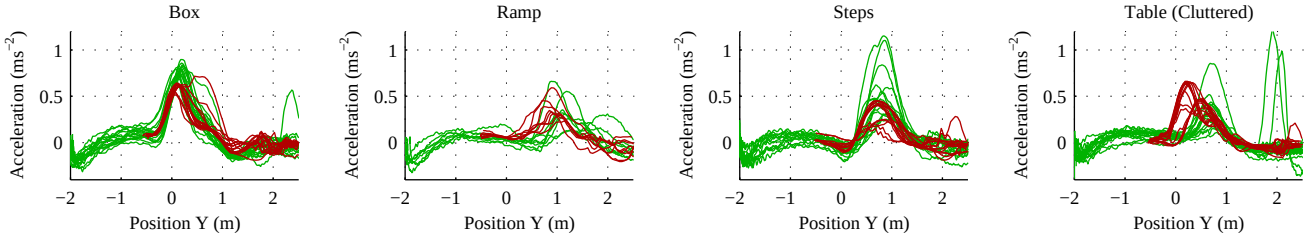

Fig. 5. Predictor output (red) against measured acceleration (green) for four example obstacles. The 'box', 'ramp' and 'steps' examples are part of the training set (which also includes six other unshown obstacle configurations including the 'no obstacle' case). The 'table (cluttered)' and 'table (empty)' (not shown) configurations are not in the training set. These flights were performed with constant thrust during the pass over the obstacle.

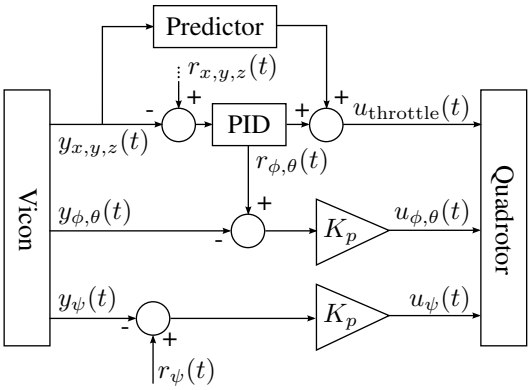

Fig. 6. Control system block diagram including the predictor stage.

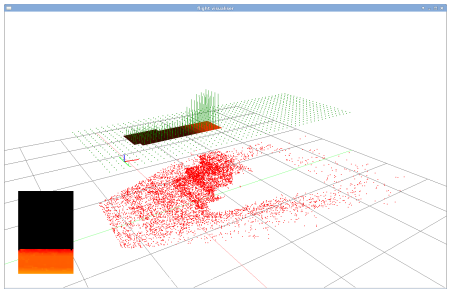

Fig. 7. A visualisation of the point cloud and prediction map. The point cloud is shown in red (this is the 'steps' obstacle). In the lower left is the synthesised depth view from the MAV's location. The MAV's current location is marked by the axis indicator. The grid of green points is the prediction map; each point represents one sample location, and the vertical lines extending from those points indicate the predicted vertical acceleration stored in the map for that location.

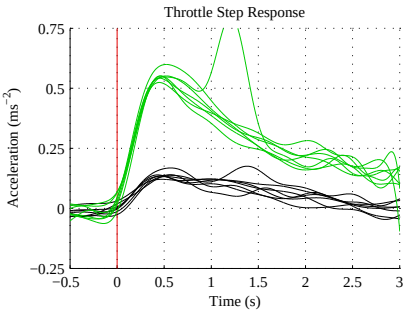

Fig. 8. Example throttle step responses for step sizes of 5 (black) and 20 (green) throttle units.

obstacle, in order to allow for differences in initial conditions, and other sources of variability. The training set we use for these experiments contains 4426 example image/acceleration pairs, and covers nine obstacle configurations (including the 'no obstacle' case).

V. Incorporating Predicted Effects into Control

In order to use the predictor to improve control of the MAV, we need to be able to make a prediction for the MAV's immediate vicinity, and adjust the control signal to compensate for the predicted effect. In our case, we are only looking at the vertical acceleration produced by the obstacle, and only the throttle signal will be adjusted to compensate.

A. Prediction Map

The predictor as described in section IV makes a prediction for a region of space 1.5 m ahead of the MAV's current location. This is too far ahead to be used directly as an input to the control loop, and so we introduce a **dynamically** updated 'prediction map' into our system. This prediction map stores predictions made over the course of the flight so that a prediction can be quickly looked up for any location seen so far. In our experiments, this prediction map is a two dimensional grid of predictions covering the flight area, with samples spaced at intervals of 10 cm.

Predictions are made continuously while the platform is performing a pass over the obstacle, with the rate limited by the computational resources available to use for prediction. Each prediction is made using the latest location of the MAV as measured by the Vicon system.

Figure 7 shows an example of the state of the prediction map part way through a trial flight. On the laptop running these experiments (with a dated 2.53 GHz Intel T9400 processor from circa 2008), predictions are made at 12–20 Hz (corresponding with a prediction time of approximately 83–50 ms).

B. Throttle Adjustment

In order to compensate for predicted obstacle disturbance acceleration, we extend the control system by adding an adjustment term to the throttle signal. The adjustment to add is computed by applying a proportional gain to a predicted acceleration for a location slightly ahead of the MAV's current position. This location offset is needed to compensate for the lag in throttle response. Since the lag in

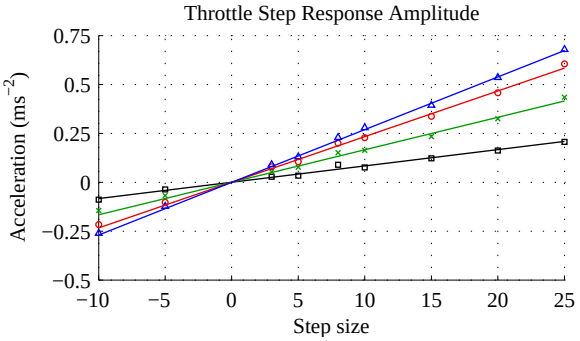

Fig. 9. Linear regression of thrust response at four different lag times (square: 0.1 s, cross: 0.2 s, circle: 0.3 s and triangle: 0.4 s), and the least-squares linear regression for each one.

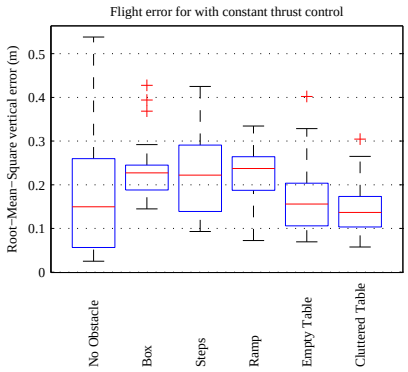

Fig. 11. Error distribution for each obstacle when the throttle is held constant. Variation largely comes from variation in initial conditions. In practice, several obstacles actually act to constrain the flight of the vehicle in this context, thereby reducing the error range (*e.g.*, compare the range for the 'no-obstacle' case against that for the 'box' case).

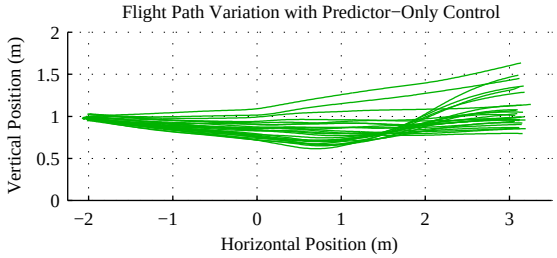

Fig. 12. Flight path variation over the 'box' obstacle, using only the predictor.

throttle response represents an offset in time, not position, the query location within the prediction map is determined by extrapolating from the MAV's location based on its current velocity, not by adding a fixed position offset.

In order to determine the gain to apply to the predicted acceleration, and the time offset to apply to the MAV's location, we attempt to characterise the thrust behaviour of the MAV. This can be done in a basic way by examining the step response of the system: commanding the MAV to hover at a particular location with a constant throttle value, and then applying an instantaneous step in throttle signal and measuring the resulting acceleration. An example plot of two such step tests with different step sizes is shown in figure 8. A reasonable initial time offset can be found by looking at the rise time of the system.

While it would be possible to fit a transfer function to the measured step responses and work from that to determine the gain factor to apply to the predicted acceleration, other factors (*e.g.*, differences in throttle response between hovering and forward flight) mean that such gain is still likely to be sub-optimal. In practice, we found it easier to perform a simpler analysis of the step response data to find a suitable range for the gain value, and then select the gain experimentally.

Figure 9 illustrates this simpler method of analysis, which is to take the response amplitude at a fixed time offset for each measured step size, and use linear regression to determine the relation between step size and acceleration. This is done for multiple time offsets during the rise period of the response in order to find a suitable range of values, within which full trials over obstacles using the predictor in the loop can be used to find an appropriate gain.

VI. Predictive Control Results

For these experiments, we have collected training data for a set of nine obstacle configurations: The 'no obstacle' case, a ramp at three different angles of inclination, flying adjacent to a wall, flying over a box, flying along the edge of a box (so that one half of the quadrotor is over the box and the other half is not), and 'steps'.

To test the predictor in the control loop, we used a set of six obstacle configurations: four (the 'no-obstacle' case, the 'box', the 'steps' and the 'ramp') which were in the training set, and two (the 'table (empty)' and 'table (cluttered)' obstacles) which were not in the training set.

We test our predictive control method by comparison to the basic (feedback only) controller described in section III-B. In order to quantify the relative performance of different control systems, we use root-mean-square vertical deviation from the demanded trajectory. This is chosen on the basis that a brief but large deviation should be penalised more than a longer but smaller deviation, since large deviations are more likely to result in a crash when flying in close proximity to obstacles.

Another possibility would be to examine the behaviour of a control system that uses only prediction, with no feedback. This is illustrated in figure 12, which shows the measured flight paths for several passes over the 'box' obstacle, using only the predictor to control the throttle channel. This example shows very high variability in the flight path, which makes it difficult to judge whether the predictor is having any useful effect. In practice, a feedback control system is required to prevent the accumulation of small errors over time, and so we think it is more adequate and realistic to

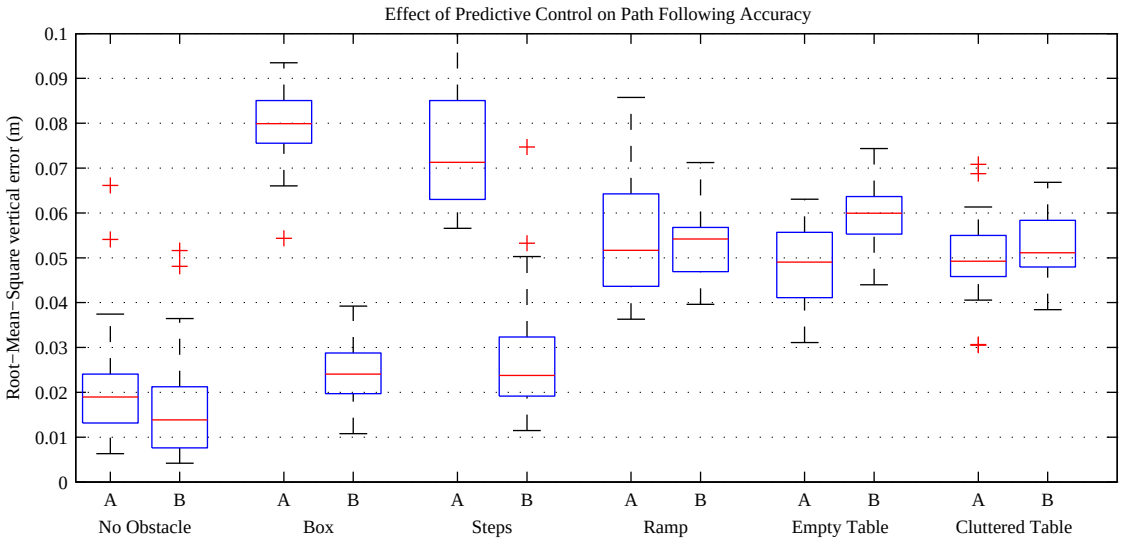

Fig. 10. Effect of predictor on error, for different obstacles. Error is measured as the root-mean-square vertical deviation from the target flight path while passing over the obstacle. Group 'A' shows the error when using normal feedback based control. Group 'B' shows the error when using feedback augmented with a prediction to compensate for the effect of the obstacle. For the 'box' and 'steps' cases, there is a statistically significant difference ($p << 0.001$) between using feedback-only and using feedback-plus-prediction, and incorporating the prediction term leads to an obvious reduction in error. For all other cases, the difference between feedback-only and feedback-plus-prediction is not statistically significant

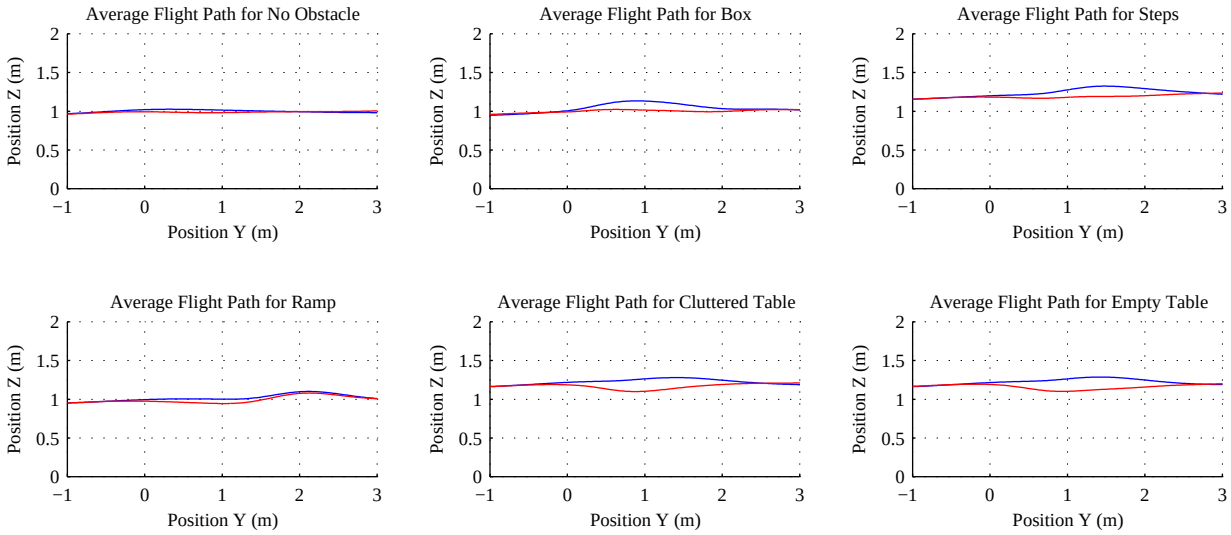

Fig. 13. Average flight path when using only feedback (blue) or using feedback-plus-prediction (red). In all cases, the demanded trajectory is flat.

test the predictor as an addition to feedback based control, rather than a replacement.

Figure 11 shows the error distribution for different obstacles when the throttle signal is held constant. This gives an indication of the range of error that can be expected with no controller.

Figure 13 shows, for each obstacle, the average flight path when using the feedback controller and the average flight path when using feedback-plus-prediction. The demanded flight path is always horizontal, so flight paths that are closer to horizontal are better. Note that the difference in flight altitude between different obstacles is deliberate; the demanded altitude is chosen to be a fixed height above the top of the obstacle, and the obstacles themselves are not all the same height.

For a quantitative comparison, figure 10 shows the distribution of error values for each obstacle, for both controllers. Of the six obstacle configurations in the test set, four of them show a small difference in error between the two controllers, while two of them (the 'box' and 'steps' obstacles) show a large, and statistically significant, reduction in error when the prediction is used.

In the 'no-obstacle' case, the similarity in error between the two controllers is expected: without an obstacle to disturb the flight, the static flight model for which the basic feedback controller is designed is reasonably accurate. In the 'ramp' case, it is not obvious why the predictor has not properly compensated for the disturbance. However, the acceleration profile for the 'ramp' obstacle (in figure 5) does show variation in both the magnitude and location of the aerodynamic effect, which reduces the effectiveness of the predictor.

Unlike the other obstacles, the two 'table' configurations were not included in the training set. Examining figure 13 again, we note that there is a clear difference between the flight paths with the two controllers. When only feedback is used, the MAV encounters a slight upward disturbance over the obstacle, but when the prediction is added, the MAV actually drops towards the obstacle. This indicates that the predicted acceleration is too large, and is predicted to occur at the leading edge of the obstacle, earlier than the true effect. The potential for making erroneous predictions when faced with an obstacle that has not previously been observed clearly warrants further attention, but one direction we can exploit with our current framework is to incorporate some notion of certainty or confidence of the predictor's output when faced with obstacles too dissimilar to objects previously experienced.

VII. Conclusions and Future Work

Feedback based control is fundamentally limited in that it can only react to disturbances after they happen. Using more sophisticated models of MAV flight dynamics can improve the control system, but cannot compensate for variation in dynamics caused by proximity to obstacles. Therefore, predicting such effects has the potential to improve control accuracy even for state-of-the-art control methods.

We have shown that using low resolution depth images (of only 40×60 pixels), learning-based prediction can be used in MAV control and that even a fairly unsophisticated prediction system can, for some commonly shaped objects, significantly improve accuracy of path following (compared to using feedback without any predictive component).

However, our method does exhibit some limitations which we believe warrant further investigation. In particular, generalisation from training examples to make predictions about very dissimilar or fully unknown environments is important, and we believe there is a lot of scope for improvement in this area. In part this can be improved from either extended training, perhaps even learning from an off-line fluid simulator, but also the incorporation of either material classification or semantic recognition of object identities could help to inform the MAV of what to do when going over an identified object.

So far, we have only examined the effects on a single axis (vertical acceleration and throttle control), but the predictions could be very easily extended to all three translation axes, and angular acceleration too.

There is also scope for predictions of the sort described in this paper to be used for path planning or other decision making at a higher level than the flight controller.

Overall, we see much scope for more work on learning from visual sensing for MAVs, which is one step beyond the purely geometric representations commonly used to date.

Acknowledgements

This work was partially funded by the UK EPSRC. Thanks to Colin Greatwood for general advice and help in using the BRL Flying Arena.

References

[1] J. Bartholomew, A. Calway, and W. Mayol-Cuevas. Learning to Predict Obstacle Aerodynamics from Depth Images for Micro Air Vehicles. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2014.

[2] Kenan Cole and Adam Wickenheiser. Impact of wind disturbances on vehicle station keeping and trajectory following. In *AIAA Guidance, Navigation, and Control (GNC) Conference*. American Institute of Aeronautics and Astronautics, 2013.

[3] I.D.Cowling, S.R. Dudhe, E.Shapiro, S.Willcox, Y.Patel, and P. Smith. Autonomous uav flight in the windy urban environment. In *24th Bristol UAV conference*, 2009.

[4] W Johnson. *Helicopter Theory*. Princeton University Press, 1980.

[5] D. Mellinger and V. Kumar. Minimum Snap Trajectory Generation and Control for Quadrotors. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, May 2011.

[6] D. Mellinger, N. Michael, and V. Kumar. Trajectory Generation and Control for Precise Aggressive Maneuvers with Quadrotors. In *Proceedings of the International Symposium on Experimental Robotics*, Dec 2010.

[7] Allen Michael and Lin Victor. Guidance and control of an autonomous soaring uav. Technical Report NASA/TM-2007-214611/REV1, NASA Dryden Flight Research Center, 2007.

[8] E.A. Nadaraya. On estimating regression. *Theory of Probability & Its Applications*, 9(1):141–142, 1964.

[9] K. Nonaka and H. Sugizaki. Integral sliding mode altitude control for a small model helicopter with ground effect compensation. In *American Control Conference (ACC), 2011*, pages 202–207, 2011.

[10] C. Powers, D. Mellinger, A. Kushleyev, B. Kothmann, and V. Kumar. Influence of aerodynamics and proximity effects in quadrotor flight. In *Experimental Robotics*, volume 88, pages 289–302. 2013.

[11] T.K. Roy and A.A. Suman. Adaptive backstepping controller for altitude control of a small scale helicopter by considering the ground effect compensation. In *Informatics, Electronics Vision (ICIEV), 2013 International Conference on*, pages 1–5, 2013.

[12] Geoffrey S. Watson. Smooth regression analysis. *Sankhy: The Indian Journal of Statistics, Series A (1961-2002)*, 26(4):359–372, 1964.