Programming Autonomous Robots: Robotics Project

RMIT University

Semester 1 – 2024

Name	Student No.	Contribution
Matthew Ricci	s3785111	33%
Noah Solman	s3285661	33%
Bailey Vogt	s3906263	33%

Contents **1**

1 Introduction

Mapping and surveying, environmental monitoring, delivery, search and rescue are some of the current applications of Unmanned Arial Vehicles (UAVs). Improving their ability to adjust to changes in wind conditions is a topic of much research (1) especially as work continues to allow UAVs to carry out increasingly complicated tasks autonomously like CSIRO's Hovermap an arial 3D mapping platform (2).

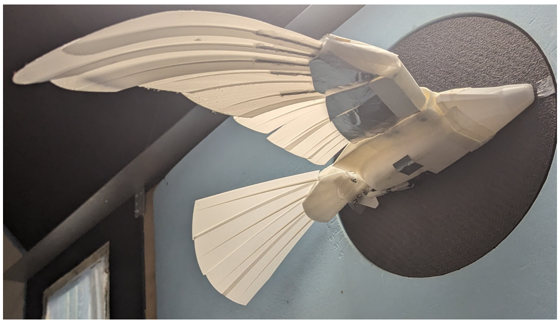


Figure 1: Kestrel Bird System

In Australia the height limit to operate a UAV is 120m above ground level (3). At these heights wind turbulence and gust conditions greatly impact the performance of UAVs, by changing its trajectory and expected current state can potentially leading to drone damage. As a response to this, researchers have turned to nature as inspiration. Basing UAV design on birds has advantages over classical UAV designs (like multi-rotor or fixed wing drones) including increased aerodynamic efficiency, high manoeuvrability and stability in high-gust conditions (4).

In this project we build on existing work to develop stable flight controllers for a novel platform the Kestrel robotic replica half-wing based on the Nankeen Kestrel with three degrees of freedom (wing extension, tail spread and tail pitch) (5) (Figure 1). As a novel design the wing cannot be controlled with existing software, prior work on the Kestrel half-wing focussed on utilising Deep Reinforcement Learning (DRL) to develop a flight stability controller. DRL has been shown to have superior response time, reduced error and ability to reject external disturbances. Additionally, development of a DRL controller does not require domain specific knowledge in controller tuning (6).

We will investigate the use of a classical controllers on the novel Kestrel platform to directly compare it to the DRL controller. We aim develop a Proportional Integral Derivative (PID) controller for flight stability already shown to be effective in stabilising UAVs (7).

2 Related Work

PID control systems have been extensively explored in UAV stabilisation. According to a 2023 survey, Lopez-Sanchez et al. conducted an exhaustive literature review and realized that the most common control technique for quadrotor UAVs is PID control (7). With adaptive PID control being the most stable under unstable conditions, providing robustness against parameter uncertainty. An adaptive PID changes the gains of the controller during flight based on performance evaluation loop built into the controller. As all our testing will be conducted in laminar flow, an adaptive PID will not be implemented and a classical PID will be developed to reduce complexity of implementation.

Tuning a PID using trial-and-error is a manual, tedious process often requiring expert domain knowledge (6). To address this, researchers are applying machine learning to the problem of finding high-performing gains for PID controllers. Support Vector Regression (SVR) is increasingly being used for PID controllers to facilitate the tuning process. Studies have shown that SVR-based controllers display enhanced stability and accuracy in variety of applications (8).

In terms of applications in bio-inspired wings, Wenfu et al.(9) in a 2021 experiment showed the effectiveness of implementing a PID in a robotic bird. They describe that as their bird gets more complex, increasingly complex methods of control will be required.

As Okasha et al. discussed in their 2022 paper. Model Predictive Control (MPC) is a powerful controller that is more robust and stable compared to PID, though it is computationally expensive (10). Another advantage over PID is that it bypasses the tuning phase.

The existing literature on PID, SVR tuning and MPC in the UAV space provided valuable insights for our project. By learning from these approaches, we can utilise a variety of control methods in a novel platform the Kestrel bio-inspired wing.

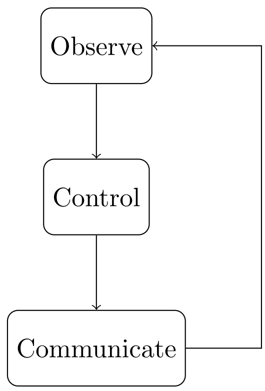
3 Methodology

3.1 System architecture

text

3.2 Software architecture

For our software system, we developed an N-tier software architecture. Our architecture contains 3 main tiers.



Firstly we have the observation tier, which links into the Kestrel's observation space, this includes the Kestrel's motor positions and collects data from the load cell that the bird is plugged into.

Secondly is the controller layer, being a component layer that allows us to plug in any of our developed controllers either the PID or the MPC controller this layer also includes the training of these controllers for example with the PID controller the SVR Tuner we have developed also runs with the data collected from the observation tier.

Finally, we have the Communication tier, which takes the place of the presentation layer. This takes everything that the previous tiers have done and sends the data to the Kestrel's action space, this then finally loops back to the observation tier to do it all again.

3.3 Developed Software

PID controller Firstly we developed a generic PID controller as a method of controlling the Kestrel bird. The PID controller takes in an initial P, I, and D value to dictate the controller's response, as well as a setpoint which is a targeted system value. For the PID controller to operate effectively the P, I and D values need to be optimally selected and for this, we move on to the next piece of software.

SVR Tuner We developed the SVR Tuner as a modern method of selecting the most optimal P, I, and D values for the PID controller, we chose this regressional method based on the identifiable success that others have had in similar applications (11). Our tuning is split into several episodes. For each episode, the tuner goes through an exploration phase to build up the data for the tuner to work with and all this data is then collected and processed to find the best PID values for our controller.

MPC We have also implemented a Model Predictive Controller, though due to the relative complexity of the algorithm, we have opted to implement a pre-existing library pyMPC which was selected due to its simplicity and the overall control we have over the setup. We chose this to compare directly against the PID controller, MPC by design is a predictive controller whereas PID is reactive, which gives us a good benchmark for the control system of the bird.

3.4 Experiment details

Wind tunnel The Kestrel will be mounted in the blue wind tunnel at the RMIT Bundoora Campus. It is a closed return wind tunnel (12). The wind tunnel is built across two floors with wind generated by a fan on the first floor sent to the second floor where the testing area is. This design ensures steady delivery of laminar flow, wind flow without disturbances. Figure 2 from displays this generic design. The vanes on each corner of the tunnel ensure that the flow is laminar.

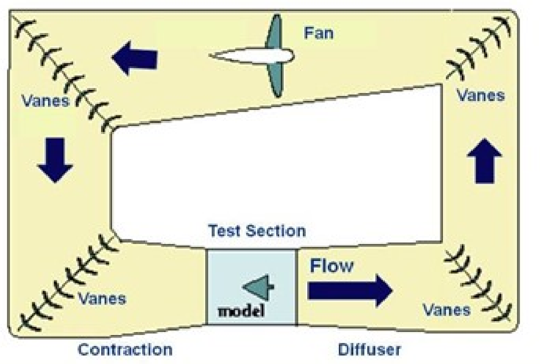


Figure 2: Closed Return Wind Tunnel (10)

A pressure probe embedded in the test area allows us to assess the pressure in the wind tunnel. Knowing the target speed and the ambient temperature and air pressure of the wind tunnel allows us to calculate a target pressure which is what we aim for when testing. All our testing occurred at approximately 0.0155 kPa or 5 m/s, the target pressure would fluctuate throughout the day as ambient temperature and pressures changed. As such we regularly calculated the target pressure to ensure consistent speeds.

Load cell The Kestrel is mounted directly onto a 30 N load cell (Figure 3). The output from the load cell describes the major forces occurring on the Kestrel. After processing through the data acquisition box, we are given a list of 6 values: the force in N at X, Y and Z. We are also given moments in X, Y and Z, which is expressed in Nm. The primary force used in training and testing was the force at Y or the lift.

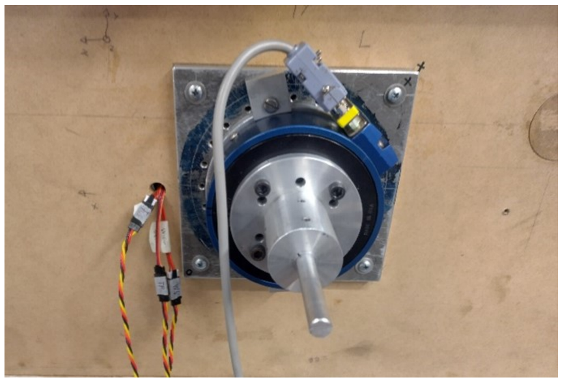


Figure 3: 30N Load Cell

Before each test the load cell must be calibrated to ensure we return accurate results. In our design we take 1000 results from the load cell and store the average for each measure in a file. This average is then subtracted from each test measurement to zero the load cell. We recognised some drift in the load cell and therefor we calibrated the load cell regularly.

Data output After a test everything that happens at each timestep this includes all the forces in N and moments in Nm, the new set position of each servo as decided by the controller and the error lift. For all tests the lift of the bird was targeted to be 0.42 N. This data will be sufficient

to capture our primary outcomes:

- Time taken to stabilise the bird
- Quality of transition to stability
- Quality of stability (error is as close to zero as possible throughout flight)

Experimental design In addition to assessing the robustness of our controllers the output will allow us to compare to the pre-trained DRL model. The following is the experimental workflow that was carried out to conduct each major test.

1. Load cell calibration, to zero the load cell

2. Dynamic pressure assessment, to ensure consistent speeds
3. Set wind speed, based on dynamic pressure assessment
4. Run test of interest PID / MPC / SVR tuner
5. Data collection

4 Results

5 Evaluation

6 Discussion

7 Conclusion

References

[1] Y. Lu and C. Liu, "Uav gust wind mitigation measurement and control system design," in *2022 IEEE International Conference on Unmanned Systems (ICUS)*, pp. 1027–1034, 2022.

[2] C. Robotics, "Hovermap."

[3] C. A. S. Authority, "Drone rules."

[4] S. Tong, Z. Weiping, M. Jiawang, and C. Zihao, "Research progress on control of bioinspired flapping-wing micro air vehicles," in *2019 IEEE International Conference on Unmanned Systems (ICUS)*, pp. 842–847, 2019.

[5] L. Stiemer, "Design of online deep reinforcement learning of servo control for a small-scale bio-inspired wind."

[6] A. Adetifa, P. Okonkwo, B. Muhammed, and D. Udekwe, "Deep reinforcement learning for aircraft longitudinal control augmentation system," *Nigerian Journal of Technology*, vol. 42, no. 1, pp. 144–151, 2023.

[7] M.-V. J. Lopez-Sanchez I, "Pid control of quadroter uavs: A survey.," 2023.

[8] K. Uçak and G. Günel, "Model-free mimo self-tuning controller based on support vector regression for nonlinear systems," *Neural Computing and Applications*, vol. 33, pp. 15731–15750, 2021.

[9] W. XU, E. PAN, J. LIU, Y. LI, and H. YUAN, "Flight control of a large-scale flapping-wing flying robotic bird: System development and flight experiment," *Chinese Journal of Aeronautics*, no. 2, pp. 235–249, 2022.

[10] M. Okasha, J. Kralev, and M. Islam, "Design and experimental comparison of pid, lqr and mpc stabilizing controllers for parrot mambo mini-drone," in *Aerospace*, no. 6 in 9, p. 298, 2022.

[11] A. Rahmani, A. Khechai, and H. Belaid, "Tuning a pd controller based on an svr for the control of a biped robot subject to external forces and slope variation," *International Journal of Advanced Robotic Systems*, vol. 11, no. 1, p. 1, 2014.

[12] NASA, "Closed return wind tunnel."