README

This document explains how to run the MATLAB code in order to reproduce the observability analysis, simulation experiments, and plotting of robotic results for the article:

**“Accommodating unobservability to control flight attitude with optic flow”**, by G.C.H.E. de Croon, J.J.G. Dupeyroux, C. De Wagter, A. Chatterjee, D.A. Olejnik, F. Ruffier., in *Nature*.

The MATLAB version used for developing the code was: MATLAB Version: 9.6.0.1072779 (R2019a). The following packages were installed (most of these are not used by the code):

Simulink Version 9.3

Aerospace Toolbox Version 3.1

Computer Vision Toolbox Version 9.0

Control System Toolbox Version 10.6

Curve Fitting Toolbox Version 3.5.9

Deep Learning Toolbox Version 12.1

Image Processing Toolbox Version 10.4

Optimization Toolbox Version 8.3

Parallel Computing Toolbox Version 7.0

Robotics System Toolbox Version 2.2

Robust Control Toolbox Version 6.6

Signal Processing Toolbox Version 8.2

Simulink Control Design Version 5.3

Statistics and Machine Learning Toolbox Version 11.5

Symbolic Math Toolbox Version 8.3

System Identification Toolbox Version 9.10

Wavelet Toolbox Version 5.2

# **All scripts**

Executing all scripts described below can be done by running the function: run\_all()

# **Nonlinear observability analysis**

In the article, we perform a nonlinear observability analysis for various models in the Supplementary Information. During our research we have implemented this analysis with the symbolic toolbox of MATLAB. Readers can regenerate all the results by running the function: observability\_analysis()

This function in turn calls functions for performing local, weak observability analysis of all studied models, local observability analysis of the elementary constant height model, and generates plots of the degree of observability, as can be found in the main article in figure 1.

# **Simulations**

The article also includes the results of simulations, which can be reproduced by running: run\_simulations()

This function in turn calls functions for determining the control performance of a model with and without rate measurements for different amounts of sensory and actuation noise (figure 1), runs the simulations used for verifying the hypothesis that attitude oscillations are reduced when flying at higher velocities (Extended Data figure 2), and runs a function that investigates the effect of active oscillations on observability (figure 3).

*Important note:* The MATLAB scripts for the simulations set a seed for the random number generator. Such a seed was not always used for the experimental results reported in the article. Consequently, although all trends are similar, the exact numbers for some of the simulation experiments are slightly different from those shown in the article.

# **Analyzing robotic experiments**

Finally, we have performed various robotic experiments for the article. The flying robots themselves have been programmed with different code bases, but we have programmed the scripts for analysis in MATLAB. All the plots with robotic results can be reproduced by running the function: analyze\_robotic\_experiments()

The function plots the experiments with the Parrot Bebop 2 quadrotor drone (figure 2, extended data figures) and with the Flapper drone (figure 3).