UAV Relay-based Auto-tuning

Daniel de Schiffart

Abstract—This thesis presents the viability of an automatic tuner of a fixed-wing UAV PID-based controller using the relay feedback method. This method, commonly applied in industrial processes, can be used to tune individual PID control loops of an aircraft of any kind, even during a flight, without little to no input from the pilot. In this thesis, a mathematical model of a fixed wing UAV is developed in parallel with a virtual relay auto-tuner, both implemented in Simulink, and then tested together to determine the ideal values for a PID controller to be implemented on said UAV.

Index Terms—Scaled flight testing, PID controlled flight, fixed-wing UAV, relay feedback, auto-tuning

I. Introduction

Scaled flight testing is an important tool in the validation of full-scale aircraft design with far lower risks and costs involved. Often full-scale aircraft are scaled down to small unmanned aerial vehicles (UAVs) with similar aerodynamic properties and tested in wind tunnels. However, with a proper control system, the UAV can be flown in open air where the measured effects are much more realistic.

Control systems for small fixed-wing UAVs are often composed of proportional-integral-derivative (PID) controllers, a simple but effective control method of low cost and complexity that fits the purpose of scaled flight testing UAVs. However, tuning these controllers is done manually in most cases, requiring experienced personnel and wasting time and resources in the process.

This thesis suggests the use of relay-based automatic tuning method suggested by Åström and Hägglund [1] to fix this problem and streamline the tuning process, making sure all the correct flying and handling requirements are met for an effective automatic tuning procedure.

A. Thesis structure

To attain its objective, the project was divided into two separate components to be developed in a simulated environment. The first is a simulation of a relay-based auto-tuner for any generic single-input, single-output (SISO) process, and the second is the development of a mathematical model of a generic UAV for use with the first component to test its performance in UAV tuning.

These components are first presented in theory, to be afterwards implemented separately in Simulink®. The MyTwinDream (MTD) UAV is used as a dataset to implement in the generic UAV model. This implementation is then used alongside the relay auto-tuner to test the ideas developed here.

II. Mathematical UAV Model

To test the implementation of an auto-tuned UAV, a mathematical model of its behaviour has to be made to simulate it in a test environment. The model receives the UAV control surface commands of its elevator, ailerons and rudder deflections and thrust throttle, seen in fig. 1 represented by δ_e, δ_a, δ_h and δ_t respectively, as inputs and returns the UAV state at any given time, after which the controller can be inserted to manipulate the control surfaces in order to keep the UAV flying steadily.

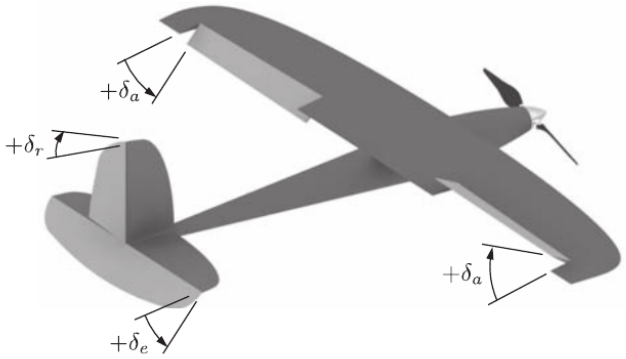

Fig. 1. Fixed-wing UAV used as a reference.

The state variables returned by the model are the positions from a north-east-down Earth-fixed frame of reference p_n, p_e, and p_d, the velocities along the body axis u, v, and w, the Euler angles of roll ϕ, pitch θ, and yaw ψ, and the angular rates p, q, and r.

A. Equations of motion

The equations of motion formulate the UAV's translation and rotation in the three-dimensional space in which it is moves by returning the derivatives of its state variables based on its current state. For a generic fixed-wing UAV, the equations defined for the aforementioned state variables can be given by [3], for the positions

$$\begin{aligned}\dot{p}_n = (c_\theta c_\psi)u + (s_\phi s_\theta c_\psi - c_\phi s_\psi)v \\ + (c_\phi s_\theta c_\psi + s_\phi s_\psi)w\end{aligned} \tag{1}$$

$$\begin{aligned}\dot{p}_e = (c_\theta s_\psi)u + (s_\phi s_\theta s_\psi + c_\phi c_\psi)v \\ + (c_\phi s_\theta s_\psi - s_\phi c_\psi)w\end{aligned} \tag{2}$$

$$\dot{p}_d = -us_\theta + vs_\phi c_\theta + wc_\phi c_\theta\,, \tag{3}$$

where c_θ represents the cosine and s_θ the sine of a variable θ, the velocities

$$\dot{u} = rv - qw - g\sin\theta + \frac{\rho V_a^2 S}{2m} C_X \tag{4}$$

$$\dot{v} = pw - ru + g\cos\theta\sin\phi + \frac{\rho V_a^2 S}{2m} C_Y \tag{5}$$

$$\dot{w} = qu - pv + g\cos\theta\cos\phi + \frac{\rho V_a^2 S}{2m} C_Z\,, \tag{6}$$

the Euler angles

$$\dot{\phi} = p + q\sin\phi\tan\theta + r\cos\phi\tan\theta \tag{7}$$

$$\dot{\theta} = q\cos\phi - r\sin\phi \tag{8}$$

$$\dot{\psi} = q\sin\phi\sec\theta + r\cos\phi\sec\theta\,, \tag{9}$$

and the angular rates

$$\dot{p} = \Gamma_1 pq - \Gamma_2 qr + \frac{1}{2}\rho V_a^2 S b C_p \tag{10}$$

$$\dot{q} = \Gamma_5 pr - \Gamma_6(p^2 - r^2) + \frac{\rho V_a^2 S\bar{c}}{2J_y} C_q \tag{11}$$

$$\dot{r} = \Gamma_7 pq - \Gamma_1 qr + \frac{1}{2}\rho V_a^2 S b C_r\,, \tag{12}$$

where the Γ factors are dependent on the mass properties of the UAV by the relations

$$\begin{aligned}
\Gamma_1 &= \frac{J_{xz}(J_x - J_y + J_z)}{J_x J_z - J_{xz}^2} & \Gamma_2 &= \frac{J_z(J_z - J_y) + J_{xz}^2}{J_x J_z - J_{xz}^2} \\
\Gamma_3 &= \frac{J_z}{J_x J_z - J_{xz}^2} & \Gamma_4 &= \frac{J_{xz}}{J_x J_z - J_{xz}^2} \\
\Gamma_5 &= \frac{J_z - J_x}{J_y} & \Gamma_6 &= \frac{J_{xz}}{J_y} \\
\Gamma_7 &= \frac{(J_x - J_y)J_x + J_{xz}^2}{J_x J_z - J_{xz}^2} & \Gamma_8 &= \frac{J_x}{J_x J_z - J_{xz}^2}
\end{aligned}\,, \tag{13}$$

and where the non-dimensional coefficients C_X, C_Y, C_Z, C_p, C_q, and C_r are defined by

$$C_X = C_X(\alpha) + C_{X_q}\frac{\bar{c}}{2V_a} q + C_{X_{\delta_e}} \tag{14}$$

$$C_Y = C_{Y_0} + C_{Y_\beta}\beta + C_{Y_p}\frac{b}{2V_a} p + C_{Y_r}\frac{b}{2V_a} r + C_{Y_{\delta_a}}\delta_a + C_{Y_{\delta_r}}\delta_r \tag{15}$$

$$C_Z = C_Z(\alpha) + C_{Z_q}\frac{\bar{c}}{2V_a} q + C_{Z_{\delta_e}} \tag{16}$$

$$C_l = C_{l_0} + C_{l_\beta}\beta + C_{l_p}\frac{b}{2V_a} p + C_{l_r}\frac{b}{2V_a} r + C_{l_{\delta_a}}\delta_a + C_{l_{\delta_r}}\delta_r \tag{17}$$

$$C_m = C_{m_0} + C_m(\alpha) + C_{m_q}\frac{\bar{c}}{2V_a} q + C_{m_{\delta_e}}\delta_e \tag{18}$$

$$C_n = C_{n_0} + C_{n_\beta}\beta + C_{n_p}\frac{b}{2V_a} p + C_{n_r}\frac{b}{2V_a} r + C_{n_{\delta_a}}\delta_a + C_{n_{\delta_r}}\delta_r\,. \tag{19}$$

B. Linearizing the model

The equations lined previously allow for the development of a non-linear model on a complex simulation platform such as Simulink® and allow for running tests on it, namely the relay feedback experiment, to analyse its behaviour. However, to perform several different control analyses on the UAV, a lower order model is often desired. For this effect, the non-linear model can be linearized around a given trim point and studied around said point, with the restriction that any analyses done in this state will be valid for a given range around it.

A non-linear model of the UAV can be summarized in an equation as

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u})\,. \tag{20}$$

If this model is linearized around a trim point, or a point where the system is in equilibrium, where the condition

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}) = 0$$

proves true, the linearized version of the equations of motion can be written as

$$\dot{\boldsymbol{x}} = \frac{\partial f(\boldsymbol{x}, \boldsymbol{u})}{\partial \boldsymbol{x}}\boldsymbol{x} + \frac{\partial f(\boldsymbol{x}, \boldsymbol{u})}{\partial \boldsymbol{u}}\boldsymbol{u}\,. \tag{21}$$

In state-space form, this can be written as

$$\dot{\boldsymbol{x}} = A\boldsymbol{x} + B\boldsymbol{u}\,. \tag{22}$$

For the case of a UAV studied using the equations of motion, the state $\boldsymbol{x}$ would look like

$$\boldsymbol{x} = \begin{bmatrix} p_n & p_e & p_d & u & v & w & \phi & \theta & \psi & p & q & r \end{bmatrix}^\intercal\,, \tag{23}$$

and the input vector $\boldsymbol{u}$

$$\boldsymbol{u} = \begin{bmatrix} \delta_e & \delta_a & \delta_r & \delta_t \end{bmatrix}^\intercal\,. \tag{24}$$

The linearized model of the UAV is obtained using a Simulink® model based on the equations of motion, and the Simulink® function trim.

III. Controller Structure

The controller was designed based on the requirements of the MicroPilot MP2128$^{\text{LRC2}}$ autopilot on which the overarching NLR project is inserted. This autopilot contains several PID loops of increasing abstraction following a scheme called loop closure. For the focus of this thesis, only the loops at the lowest level of this controller design will be considered.

Another thing to take into account before we present the loops are gain schedules. The way the MP2128$^{\text{LRC2}}$ is structured it supports four different gain schedules per loop, which toggle based on the current flight phase. Since the UAV model developed in this thesis is linearized around steady flight, this flight condition will be in focus, meaning its corresponding gain schedule is used.

With this in mind, out of the four control inputs δ_e, δ_a, δ_r, and δ_t presented,

- δ_e will be defined based on the desired pitch angle θ;
- δ_a will be defined based on the desired roll angle ϕ;
- δ_r will be defined based on the desired Y acceleration $\dot{v}$ (zero);
- δ_t will be fixed.

The controller requirements are defined by the RPV Flying Qualities Design Criteria [11] technical report, which defines the preferred flying and handling qualities for an unmanned aircraft, whether UAV or remotely piloted vehicle (RPV).

The cited document defines four categories of flight phases.

- Category A: Flight phases that require precision and rapid movement.
- Category B: Flight phases with slow, gradual movements and less precision required.
- Category C: Launch/recovery flight phases with similar requirements to category A.
- Category D: Launch/recovery flight phases with similar requirements to category B.

Since we're analysing the UAV and tuner's performance in level flight, category B will be considered.

There are multiple types of uses for UAV-based missions, which affect how the UAV has to be handled. Prosser and Wiler gathered a list of these uses.

- Class I: Small, light mini UAVs, such as surveillance, reconnaissance, targets, demonstration models, electronic warfare or harassment models;
- Class II: Low-maneuverability UAVs, such as those used in surveillance, reconnaissance, or other long endurance missions at high altitudes;
- Class III: Medium-maneuverability UAVs, similarly used in surveillance, reconnaissance and similar at low altitudes, low-level terrain following and avoidance missions, weapon delivery, and larger target UAVs;
- Class IV: High-maneuverability UAVs, such as interceptors and air-to-air combat targets.

Considering this description of UAV classes, and considering the mission for this project, the MTD model used falls into the Class I category, and this category will be used for any discussions further on.

The quality of a given UAV's flight is separated into three levels with different characteristics for each analysed flying property. These levels share the analysed property's ability to fulfill the goals for which the UAV is designed. Ideally, the UAV should aim for level 1 in all conditions.

In case of autonomous control, the three levels are classified as follows.

- Level 1A: UAV flying qualities are adequate to perform their corresponding mission phase.
- Level 2A: UAV flying qualities are adequate to perform their corresponding mission phase with some degradation of mission effectiveness.
- Level 3A: UAV flying qualities are adequate to recover the vehicle.

For manual control in a RPV mode, the three levels are described as such.

- Level 1M: Remote control is adequate to perform the corresponding mission phase.
- Level 2M: Remote control is adequate to perform the corresponding mission phase, with an increase in operator work load and decrease in mission effectiveness.
- Level 3M: Remote control is degraded but still adequate to recover vehicle.

For most cases, including the current case, the separation into automatic and manual flying qualities is not relevant and it can be assumed that the levels on both versions are analogous to each other.

Considering a class I UAV and a category B flight condition, the dynamic mode requirements for all five modes (short-period, phugoid, roll, spiral, and dutch-roll) are given by the aforementioned document [11].

The requirements for the short-period mode are given in table I and fig. 2, where the value of $\frac{n}{\alpha}$ can be given by

$$\frac{\rho V_0^2 S}{2mg} C_{L_\alpha} \,. \tag{25}$$

TABLE I
Requirements for the UAV short-period damping.

Level	Requirement
Level 1	$0.30 < \zeta_{\text{sp}} < 2$
Level 2	$0.20 < \zeta_{\text{sp}} < 2$
Level 3	$\zeta_{\text{sp}} > 0.15$

The requirements for the phugoid are given in table II. In the level 3 requirement, the time constant T_2 is given by

$$T_2 = \frac{\ln 2}{-\zeta_{\text{ph}} \omega_{n_{\text{ph}}}} \,. \tag{26}$$

TABLE II
Requirements for the UAV phugoid damping.

Level	Requirement
Level 1	$\zeta_{\text{ph}} > 0.04$
Level 2	$\zeta_{\text{ph}} > 0$
Level 3	$T_2 > 55\,\text{s}$

The requirements for the roll mode, are given in table III.

TABLE III
Requirements for the UAV roll time constant.

Level	Requirement
Level 1	$\tau_r < 1$
Level 2	$\tau_r < 3$
Level 3	$\tau_r < 10$

For the spiral mode, the requirements are measured according to time T, which is the time taken by the UAV to double its roll angle value following a disturbance of up to 20 degrees. These requirements are given in table IV

TABLE IV
Requirements for the UAV spiral mode, where T is the time taken by the UAV to double its roll angle value following a disturbance of up to 20 degrees.

Level	Requirement
Level 1	$T > 20\,\mathrm{s}$
Level 2	$T > 12\,\mathrm{s}$
Level 3	$T > 4\,\mathrm{s}$

The requirements for the dutch-roll mode are given in table V.

TABLE V
Requirements for the UAV dutch-roll properties.

Level	Requirements		
Level 1	$\zeta_{\mathrm{dr}} > 0.19$	$\omega_{n_{\mathrm{dr}}} > 1$	$\omega_{n_{\mathrm{dr}}}\zeta_{\mathrm{dr}} > 0.35$
Level 2	$\zeta_{\mathrm{dr}} > 0.02$	$\omega_{n_{\mathrm{dr}}} > 0.4$	$\omega_{n_{\mathrm{dr}}}\zeta_{\mathrm{dr}} > 0.05$
Level 3	$\zeta_{\mathrm{dr}} > 0.02$	$\omega_{n_{\mathrm{dr}}} > 0.4$	

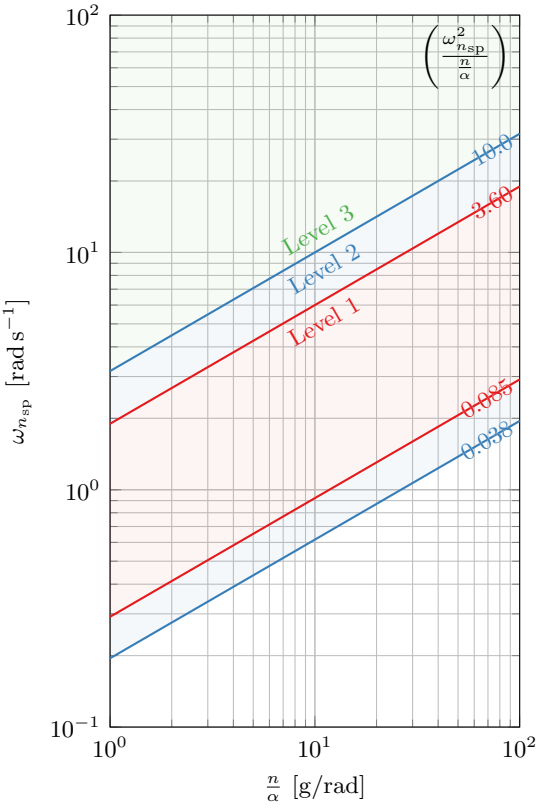

Fig. 2. Requirements for the short-period natural frequency requirements for a category B UAV. The diagonal lines on the logarithmic scale are defined by the ratio in the upper-right corner. The areas outside of the graph can be obtained by extending the defined lines. Adapted from [11, 4].

IV. Relay Auto-tuner

A relay-based auto-tuner makes use of the properties of a relay feedback experiment on a certain process to automatically determine the parameters for a PID controller that is to be applied to the same process.

When testing a feedback loop with a simple gain as a compensator, starting with a low gain and increasing it will bring a process into an oscillatory state in which the oscillations remain at a steady amplitude, nearing instability. The gain at this point is defined as the ultimate gain k_u, and the frequency of these oscillations is the ultimate frequency ω_u. Ziegler and Nichols [12] make use of this gain to define parameters for a PID controller with a good performance in a set of equations which became known as tuning rules.

A relay feedback experiment is a process in which a feedback loop with a PID controller has its controller replaced by a relay, as in fig. 3 that switches between outputs $-d$ and $+d$ depending on the value of the input. When this occurs, the process oscillates at the

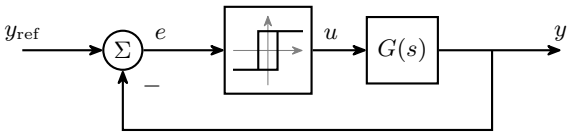

Fig. 3. Feedback loop used for the relay feedback experiment.

ultimate frequency and with a steady amplitude called the ultimate amplitude. The ultimate amplitude can be used to determine the ultimate gain in a relation defined as

$$k_u = \frac{4d}{a\pi}\,. \tag{27}$$

Many different tuning rules exist, with various different performances dependent on the process they are applied to. 44 rules were gathered from [9] and tested automatically using the linearized model, and 4 rules were chosen based on the flying and handling qualities of the closed-loop UAV after each tuning rule is applied. These are presented in table VI.

TABLE VI
Tuning rules selected for this thesis. Adapted from [9].

Author	$\boldsymbol{k_c}$	$\boldsymbol{T_i}$	$\boldsymbol{T_d}$
Pettit and Carr [10]	$0.5k_u$	$1.5T_u$	$0.167T_u$
Fuxiang and Zhixiong [5]	$0.27k_u$	$2.40T_u$	$1.32T_u$
Luyben and Luyben [7]	$0.46k_u$	$2.20T_u$	$0.16T_u$
Åström and Hägglund [2]	a_1	a_2	a_3

In table VI, the coefficients a_1, a_2, and a_3 are given by

$$a_1 = k_u \cos\phi_m \tag{28}$$

$$a_2 = \alpha a_3 \tag{29}$$

$$a_3 = \frac{\tan\phi_m + \sqrt{\frac{4}{\alpha} + \tan^2\phi_m}}{2\omega_u}, \tag{30}$$

where ϕ_m represents a desired phase margin and α an adjustable parameter, found to be 5 for the MTD UAV.

A simulated auto-tuner was developed in Simulink® for the project, based on the work of Hornsey [6].

V. MyTwinDream

The MyTwinDream is a small off-the-shelf styrofoam UAV used by the SCALAiR project to test the implementation of various features of the MicroPilot autopilot and to allow for the pilots the adapt to the hardware's implementation and characteristics. It includes two SunnySky X2814-KV900 propeller engines, one per wing, as well as two ailerons, elevators, and a single rudder for its control surfaces. A photo of the MTD can be found in fig. 4.

Fig. 4. The MyTwinDream UAV with its propeller blades removed.

The data for this UAV was used to test the project's implementation in a simulated environment. The UAV's characteristics are presented in table VII.

TABLE VII
Parameters for the MTD aircraft.

Property	Value
Mass	3.156 kg
Wing span	1.8 m
Wing area	0.4736 m
Mean aerodynamic chord	0.2637 m
J_{xx}	0.213 kg m^2
J_{yy}	0.171 kg m^2
J_{zz}	0.350 kg m^2
J_{xy}	0.04 kg m^2
J_{xz}	0.00 kg m^2
J_{yz}	0.00 kg m^2

The thrust data was obtained in multiple look-up tables obtained by PropCalc – Propeller Calculator Software. A three-dimensional plot of the data can be found in fig. 5.

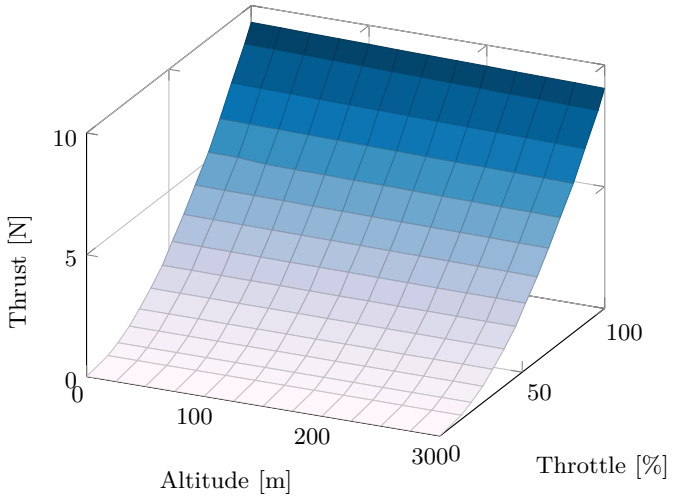

Fig. 5. SunnySky X2814-KV900 thrust force for a single engine based on throttle command and height.

The model was linearized around a cruise speed of 17 m s^{-1}, level flight, at a standard IAS ground level air density of 1.225 kg m^{-3}. The wind was considered to be still. The model's state-space model was split into two separate state-space models representing the UAV's longitudinal and lateral behaviour, to simplify analysis and representation, with the dynamics of the aircraft position p_n, p_e and p_d dropped for not being necessary for the analysis that follows. Longitudinal behaviour in this case refers to movement along the UAV's x and z axes and rotation around its y axis, while lateral behaviour considers movement along the UAV's y axis and rotation around its x and z axes.

With this in mind, the model's longitudinal state-space model is

$$\begin{bmatrix}\dot{u}\\ \dot{w}\\ \dot{q}\\ \dot{\theta}\end{bmatrix} = \begin{bmatrix}-0.05814 & 0.3051 & 0.05462 & -9.807\\ -1.182 & -7.863 & 15.13 & 0.0354\\ -0.03191 & -8.839 & -10.96 & 0\\ 0 & 0 & 1 & 0\end{bmatrix}\begin{bmatrix}u\\ w\\ q\\ \theta\end{bmatrix} + \begin{bmatrix}-0.05165 & 4.14\\ -14.31 & 0\\ -171.8 & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix}\delta_e\\ \delta_t\end{bmatrix}, \tag{31}$$

and the lateral model is

$$\begin{bmatrix}\dot{v}\\ \dot{p}\\ \dot{r}\\ \dot{\phi}\\ \dot{\psi}\end{bmatrix} = \begin{bmatrix}-0.3047 & 0.02019 & -16.78 & 9.807 & 0\\ -1.021 & -18 & 3.638 & 0 & 0\\ 1.722 & -0.5569 & -1.376 & 0 & 0\\ 0 & 1 & -0.00361 & 0 & 0\\ 0 & 0 & 1 & 0 & 0\end{bmatrix}\begin{bmatrix}v\\ p\\ r\\ \phi\\ \psi\end{bmatrix} + \begin{bmatrix}-0.7077 & 3.805\\ 259.8 & 5.683\\ 0 & -24.7\\ 0 & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix}\delta_a\\ \delta_r\end{bmatrix}. \tag{32}$$

The models obtained in eqs. (31) and (32) allow us to obtain transfer functions for the dynamics of the UAV, which are not used directly in the auto-tuning of the PID loops, but are used for analysis of the UAV's expected behaviour.

The Elevator from Pitch transfer function obtained from eq. (31) looks like

$$\frac{\theta(s)}{\delta_e(s)} = \frac{-171.85(s+7.075)(s+0.1099)}{(s^2+0.03527s+0.4542)(s^2+18.84s+220.2)}, \tag{33}$$

the Aileron from Roll transfer function from eq. (32) is

$$\frac{\phi(s)}{\delta_a(s)} = \frac{259.78s(s^2+1.686s+29.3)}{s(s+17.96)(s-0.08412)(s^2+1.809s+30.8)}, \tag{34}$$

and the Rudder from Y Acceleration transfer function also obtained from eq. (32) is

$$\frac{\dot{v}(s)}{\delta_r(s)} = \frac{3.8049s^2(s+110)(s+18.45)(s-0.1021)}{s(s+17.96)(s-0.08412)(s^2+1.809s+30.8)}. \tag{35}$$

These values can now be used to analyse the UAV's open-loop[1] flying and handling qualities. The qualities of the UAV are dependent on its dynamic modes and can be obtained from the longitudinal model, which describes the short-period and phugoid modes, and the lateral model, which describes the roll, spiral, and dutch-roll modes. The values describing these characteristics can be obtained from the eigenvalues from the $\boldsymbol{A}$ matrix from the corresponding state-space models.

For the case in eq. (31), the eigenvalues are

$$\lambda_{\text{lon}} = \begin{matrix}-0.0176 \pm 0.6737i\\ -9.422 \pm 11.46535i\end{matrix}. \tag{36}$$

The short-period eigenvalues are the complex pair $-9.422 \pm 11.46535i$ and the phugoid eigenvalues are $-0.0176 \pm 0.6737i$.

The short-period dynamics are then obtained from these eigenvalues. The short-period natural frequency $\omega_{n_{\text{sp}}}$ is determined to be $14.84\,\text{rad}\,\text{s}^{-1}$ and its damping ζ_{sp} is 0.635. Analogously, the phugoid natural frequency $\omega_{n_{\text{phu}}}$ is $0.674\,\text{rad}\,\text{s}^{-1}$ and its damping ζ_{phu} is 0.0262.

For the lateral model in eq. (32), the eigenvalues are determined to be

$$\lambda_{\text{lat}} = \begin{matrix}-17.9593\\ -0.9044 \pm 5.476i\\ 0.0841\\ 0.0\end{matrix} \tag{37}$$

In this case, the roll eigenvalue is -17.9593, the spiral eigenvalue is 0.0841 and the dutch-roll eigenvalues are the complex pair $-0.9044 \pm 5.476i$. The eigenvalue at the origin is not attributed to any mode and can thus be ignored.

The dutch-roll and roll dynamics used in the study of flying and handling qualities can thus be obtained from these results. For the dutch roll, its natural frequency $\omega_{n_{\text{dr}}}$ is $5.55\,\text{rad}\,\text{s}^{-1}$ and its damping ζ_{dr} is 0.163. For the roll dynamics, its time constant τ_{r} is found to be 0.0557 s.

The flying and handling qualities of the UAV can be determined. For category A flight phases, short-period can be seen to equate to level 3, phugoid to level 2, roll to level 1, and dutch-roll to level 2. For category B flight phases, short-period is level 3, phugoid level 2, roll level 1, and dutch-roll level 1. Spiral mode had to be tested separately in the Simulink® model of the MTD, but after giving the UAV a roll angle ϕ disturbance of 10° its value doubled to 20° after 6.615 s, giving the spiral mode a level 3. All in all, this means that the MTD can be classified as a level 3 UAV for category A and B flight phases.

VI. Experiment and Results

Once the UAV model and the auto-tuner were developed in Simulink®, a simulation environment was made for the UAV to perform simple trimmed test flights with feedback, and afterwards three relay auto-tuners were applied, one per loop to be tuned. Once applied, the tests were ran, three tests per tuning rule that were run sequentially until all three loops were tuned and trimmed.

The relay auto-tuner settings were defined per loop, but were the same for all tuning rules. Their settings were estimated based on which control surface deflections could provide a feedback strong enough without reaching their maximum values and bring the UAV to an unstable condition, while the signal for the relay output was determined based on the steady-state gain for each loop determined from their corresponding transfer functions.

The settings and results per loop can be found in table VIII. The results for each ultimate frequency and gain were the same for every tuning rule and were consolidated into one table that is valid for every rule.

The time plot of the first second of the relay feedback experiment for each loop can be found in figs. 6 to 8. These represent the values for the commanded process input in the blue line, controlled in these scenarios by a relay, and the red value represents the process output.

Various differing behaviours can be seen in these three plots, but the first analysis can be focused on the Elevator from Pitch loop, represented in fig. 6. Since the simulation

TABLE VIII
Relay feedback experiment results.

Loop	d	ϵ	ω_u	k_u
Elevator from Pitch	$-10°$	0	25.761	-3.678
Ailerons from Roll	$20°$	0	17.907	1.818
Rudder from Y Acceleration	$3°$	1 g	35.814	0.187

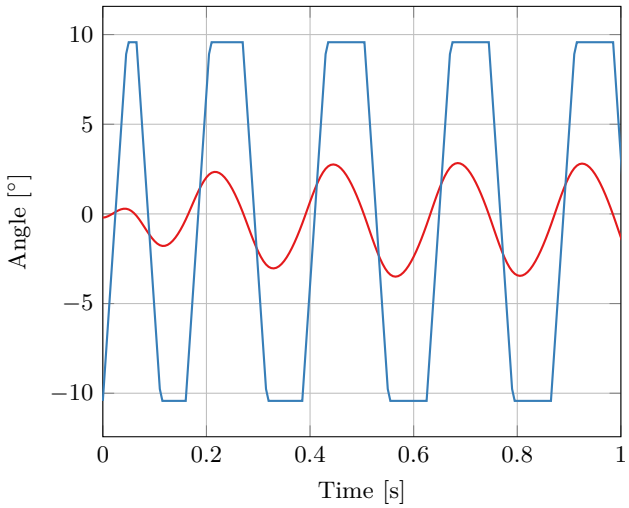

Fig. 6. Behaviour of the elevator during the simulated relay feedback experiment. The red line is the process variable, and the blue line is the commanded variable, controlled by a relay.

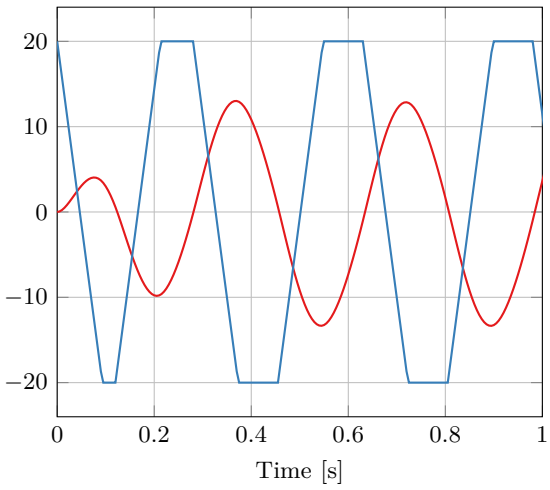

Fig. 7. Behaviour of the ailerons during the simulated relay feedback experiment.

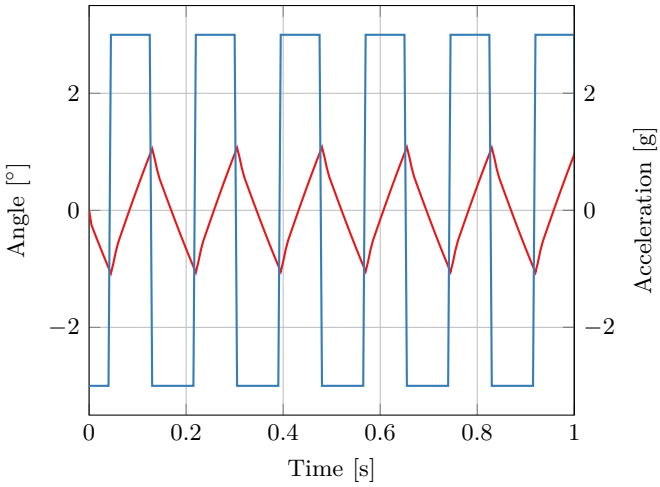

Fig. 8. Behaviour of the rudder during the simulated relay feedback experiment.

starts with the UAV in a trimmed condition, the oscillation of the pitch angle first requires just about three full oscillations to reach its full amplitude, and at this point the relay auto-tuner is ready to determine the ultimate frequency and gain. Also because of the initial trim condition, the relay is seen to oscillate not exactly between $-d$ and $+d$, but between their differences with the set point, which was calculated to be different than zero. Not only that, but its output is not a perfectly square-type wave, but rather one with a ramping of its signal around the switching points. This happens due to the implementation of actuator models, which restrict the maximum rate of change of the elevator deflection δ_e.

The behaviour of the second loop, Ailerons from Roll, visible in fig. 7, is very similar to the Elevator from Pitch, with longer and bigger oscillations than its elevator counterpart. This can be explained by the higher value of the set relay output d, which leads to these bigger oscillations. However, since the value of the outputted PID parameters depends on both amplitude and frequency, adjusting the value of d will not affect the final values of the PID parameters. In this case, a bigger value for d was chosen as it allows the experiment to obtain more conclusive results.

The last loop, Rudder from Y Acceleromter, is, in this scenario, the oddest of the three. The shape of the Y acceleration does not take a near-sinusoidal form as its other counterparts, but rather a triangular shape. Also to note is that its waveform changes direction when it reaches 1 g. All these effects are caused by the presence of a positive hysteresis ϵ of 1 g, which causes the waveform to obtain its characteristic shape. The value of the hysteresis is well visible in fig. 8, in an interval in which the value of the Y acceleration remains steady until its next point of inversion.

Once the relay feedback experiments are performed and

the results for the ultimate frequencies and gains are found, the four tuning rules can be applied. Their values are given in tables IX to XI.

TABLE IX
Tuning results for the Elevator from Pitch loop.

Author	k_c	T_i	T_d
Pettit and Carr	−1.8391	0.3658	0.0407
Fuxiang and Zhixiong	−0.9931	0.5853	0.3219
Luyben and Luyben	−1.6920	0.5365	0.0390
Åström and Hägglund	−1.8391	0.3572	0.0714

TABLE X
Tuning results for the Ailerons from Roll loop.

Author	k_c	T_i	T_d
Pettit and Carr	0.9092	0.5263	0.0585
Fuxiang and Zhixiong	0.4899	0.8421	0.4631
Luyben and Luyben	0.8365	0.7719	0.0561
Åström and Hägglund	0.9092	0.5139	0.1027

TABLE XI
Tuning results for the Rudder from Y Acceleromter loop.

Author	k_c	T_i	T_d
Pettit and Carr	0.0936	0.2631	0.0292
Fuxiang and Zhixiong	0.0502	0.4210	0.2315
Luyben and Luyben	0.0855	0.3859	0.0280
Åström and Hägglund	0.0936	0.2569	0.0513

In figs. 9 to 11 the step tests for the MTD transfer functions with PID control applied are found. The parameters tested for the PID come from different tuning rules applied to the same results from table VIII.

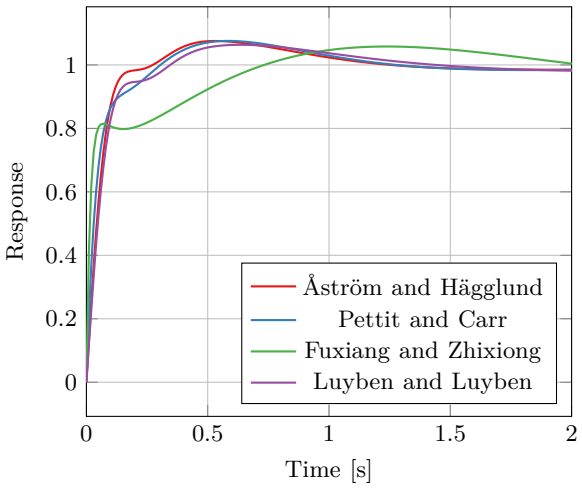

Fig. 9. Step tests for the tuned Elevator from Pitch loop.

The Elevator from Pitch loop performance, shown in fig. 6, shows quite the similarity between the step

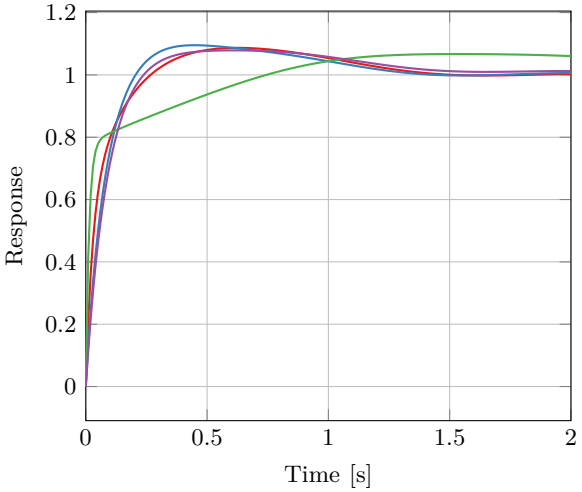

Fig. 10. Step tests for the tuned Ailerons from Roll loop.

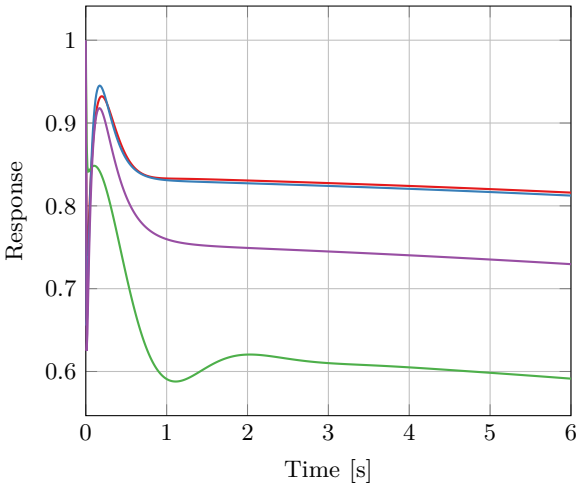

Fig. 11. Step tests for the tuned Rudder from Y Accelerometer loop.

responses obtained by each rule, with the exception being the Fuxiang and Zhixiong, which behaves quite uniquely, consequence of its increased integrative and derivative actions. It is the fastest of the rules to reach its first peak at around 0.8, and then corrects its own steady state error, but due to the increased integrative action, some slower oscillation is induced afterwards and corrected more slowly due to a reduced proportional action, making it the slowest set of parameters to converge.

Between the remaining sets of rules, Åström and Hägglund appears to provide the fastest peak and settling time but by a very small margin. The remaining rules perform appropriately and with very little distinction from this last one.

From the Ailerons from Roll loop performance, shown in fig. 7, once again the exception is the Fuxiang and Zhixiong rule, which shows the fastest peak time and damping with the smallest overshoot, but slowest settling

time, having considerable oscillations ($\simeq 0.05$) even after the two seconds displayed. The remaining parameters now have slightly bigger differences between their responses, with Pettit and Carr now having the fastest peak time but bigger overshoot in comparison to the other two.

The Rudder from Y Accelerometer provides the most interesting step response. Not only does it not reach the given target of 1, but it also slowly decays away from said target back to zero. Since the analysis is performed here using the linearized transfer functions, this slow decay cannot be caused by other effects of the UAV, but it can be attributed to integral action of the implemented PID controller, with is further evidenced by the fact that the tuning rules with more integrative action are diverging from the target 1 the fastest, as can be seen in fig. 8. This points to the fact that the PID controller assumes a steady-state error, meaning it wants to lead the input signal back to zero. As the Rudder from Y Acceleration loop is used to maintain the acceleration along the y axis zero, this does not pose as much of an issue.

With these results, it can be asserted that for the tuning rules that satisfy good flying and handling quality requirements, Fuxiang and Zhixiong provides the fastest peak, smallest overshoot, and highest damping properties, but its reliance on integrative and derivative action means its settling time is eventually sacrificed, and could pose a problem once noise models are introduced. For a more reliable alternative, between the remaining tuning rules, due to the similarity in their results, the Åström and Hägglund rule would be the preferred choice for general implementation scenarios, as it provides much the same properties but allows for the specification of a phase margin and provides an extra tuning parameter that regulates the influence of integrative and derivative action, should so be necessary.

VII. Conclusion

The mathematical model of the MTD UAV was successfully developed based on its geometrical and physical data, and a trim point was found for a level flight condition. Alongside this development, the relay auto-tuner was developed on the Simulink® platform to interact with any given process that it was implemented in, being validated every step of the way.

In being developed and validated separately, the interaction between both components eventually determined the outcome of the plan this thesis set out to accomplish, which was successful for the tests that were carried out with both platforms. Relay feedback was achieved for all input-output pairings tested for the MTD, and sets of parameters for desired PID controllers were obtained and compared to each other based on their performance. With this, the thesis showed that not only is a relay-based auto-tuner theoretically possible for small-scale PID-controlled UAV, but that it satisfies the required conditions for good flying and handling qualities and that there is room for improvement and adjustment in a case-by-case approach should more requirements come to play for different missions.

References

[1] Karl Johan Åström and Tore Hägglund. "Automatic Tuning of Simple Regulators". In: IFAC Proceedings Volumes 17.2 (1984), pp. 1867–1872. doi: 10.1016/S1474-6670(17)61248-5.

[2] Karl Johan Åström and Tore Hägglund. "Automatic Tuning of Simple Regulators with Specifications on Phase and Amplitude Margins". In: Automatica 20.5 (1984), pp. 645–651. doi: 10.1016/0005-1098(84)90014-1.

[3] Randal W. Beard and Timothy W. McLain. Small unmanned aircraft. Theory and practice. Jan. 2012. isbn: 978-0-691-14921-9. url: http://uavbook.byu.edu/doku.php (visited on 04/01/2019).

[4] Dante A. DiFranco. Flight Investigation of Longitudinal Short Period Frequency Requirements and PIO Tendencies. Tech. rep. Version AFFDL-TR-66-163. Air Force Flight Dynamics Laboratory, Wright-Patterson Air Force Base, June 1967.

[5] Chen Fuxiang and Yang Zhixiong. "Self-Tuning PM Method and its Formulas Deduction in PID Regulators". In: Zidonghua Xuebao/Acta Automatica Sinica 19.6 (1993), pp. 736–740.

[6] Stephen Hornsey. "A Review of Relay Auto-tuning Methods for the Tuning of PID-type Controllers". In: 5.2 (2012): Reinvention: an International Journal of Undergraduate Research. issn: 1755-7429. url: https://warwick.ac.uk/fac/cross_fac/iatl/reinvention/archive/volume5issue2/hornsey (visited on 06/14/2019).

[7] Michael L. Luyben and William L. Luyben. Essentials of Process Control. McGraw-Hill, 1997.

[8] Markus Müller. PropCalc – Propeller Calculator Software. Version P7.16c. Mar. 25, 2019. url: https://www.ecalc.ch/motorcalc.php (visited on 07/26/2019).

[9] Aidan O'Dwyer. Handbook of PI and PID controller tuning rules. 3rd ed. Imperial College Press, 2009. isbn: 1-84816-242-1.

[10] John W. Pettit and Douglas M. Carr. "Self-tuning controller". 4669040. May 26, 1987.

[11] Charles F. Prosser and Curtiss D. Wiler. RPV Flying Qualities Design Criteria. Tech. rep. Version AFFDL-TR-76-125. AFFDL/FGC, Dec. 1976.

[12] John G. Ziegler and Nathaniel B. Nichols. "Optimum Settings for Automatic Controllers". In: ASME 64.11 (1942), pp. 759–768.