

Aircraft Pitch Control via Static and Reinforcement Learning-Based Adaptive PID Controllers

Muhammad Alfiyandy Hariansyah, *PhD Student, HKUST*

Abstract—In this project, a static Proportional-Integral-Derivative (PID) controller and a Reinforcement Learning (RL)-based adaptive PID controller are developed to control the pitch of an aircraft system. The static PID represents a traditional method where the PID gains are constant and tuned beforehand, while the RL-based adaptive PID represents an intelligent method where the PID gains are varying over time. The project begins with deriving the non-linear dynamic system from the first principles. Subsequently, the system of equations is linearized and separated into two uncoupled motions: longitudinal and lateral. The pitch controllers are then designed based on the linearized longitudinal equations. The aircraft considered here is a Cessna-172 due to the availability of its stability derivatives. This report presents and discusses results for both static PID- and RL-based adaptive PID-controlled pitch simulations.

Index Terms—Aircraft pitch control, PID, adaptive tuning, reinforcement learning, linearization, Cessna-172

I. INTRODUCTION

IN aviation, precise control over an aircraft's attitude is critical for ensuring stability, responsiveness, and safety during flight operations. The motion of an aircraft is highly complicated, coupled, and non-linear [?], described via a 6-DOF motion: three translational motions (vertical, horizontal, transverse) and three rotational motions (roll, pitch, yaw). The control over these motions is achieved by regulating the throttle and the control surfaces: elevator, aileron, and rudder. Furthermore, the control system is separated for two uncoupled motions: longitudinal and lateral. The longitudinal motion describes the pitch and velocity attitudes controlled by the throttle and elevator. Meanwhile, the lateral motion describes the roll and yaw attitudes, controlled by aileron and rudder.

Controlling the pitch of an aircraft in longitudinal motion can be achieved by regulating the elevator and keeping the constant thrust. To reduce the complexity, the aircraft is assumed to have a rigid body, and its motion is only a small deviation from its equilibrium flight condition [?], i.e., level unaccelerated flight. In this way, the highly coupled and non-linear system of equations can be linearized and used to design our controllers. Although, recent works have developed the flexibility part in the aircraft modeling [?].

Aircraft pitch control is not a new problem. Others have developed pitch controllers based on traditional methods, such as Proportional-Integral-Derivative (PID) [?], Linear-Quadratic-Regulator (LQR) [?], and Observer-State-Feedback (OSF) [?]. These methods rely on static gains that are tuned and optimized beforehand. However, they possess a strong

static performance in a single condition they are tuned for, and their performance is reduced when exposed to time-varying uncertain factors [?]. Recent works have also attempted to use intelligent methods, such as a Fuzzy PID [?], Back-Propagation Neural Network [?], and Reinforcement Learning (RL) [?], [?], which have some self-tuning and self-optimization features of the control parameters. Tuning of control parameters can be achieved via manual tuning, optimization, or RL. Mohanty et al. [?], for example, uses a class of RL algorithm called Deep Deterministic Policy Gradient (DDPG) [?] to tune the PID controller.

Although aircraft pitch control has been extensively studied, there still remains an open challenge both in the modeling and control parts. In this project, a highly coupled and non-linear system of equations is modeled, assuming aircraft data for Cessna-172 [?], see Appendix ???. Subsequently, linearization is performed to derive the linear equations. Finally, a traditional PID control with different static gains and an RL-based adaptive PID control with different neural network architectures under Proximal Policy Optimization (PPO) [?] framework are compared and their performance is discussed.

II. NON-LINEAR COUPLED 6-DOF AIRCRAFT MOTION

The equations of motion for a flight vehicle usually are written in a body-fixed coordinate system. The body velocities are defined as: $\vec{v}_B = (u, v, w)^T$, while body angular rates are defined as: $\vec{\Omega}_B = (p, q, r)^T$. The velocities can also be written in earth-fixed coordinate system (xyz -axis) as $\vec{v}_E = (\dot{x}, \dot{y}, \dot{z})^T$. In similar manner, the roll, pitch, and yaw angles are defined as ϕ , θ , and ψ , respectively. The following transformation can be done to relate the body velocities \vec{v}_B to \vec{v}_E : $\vec{v}_E = \mathcal{L}_{B \rightarrow E} \vec{v}_B$, where $\mathcal{L}_{B \rightarrow E}$ is the body-to-earth transformation matrix, see Appendix ?? for details. By performing body-to-earth transformation on velocities, the following equations are obtained:

$$\dot{x} = u \cos \psi \cos \theta - v (\cos \phi \sin \psi + \sin \phi \sin \theta \cos \psi) + w (\sin \phi \sin \psi + \cos \phi \sin \theta \cos \psi) \quad (1)$$

$$\dot{y} = u \cos \theta \sin \psi + v (\cos \phi \cos \psi + \sin \phi \sin \theta \cos \psi) - w (\cos \phi \sin \theta \cos \psi - \sin \phi \sin \theta \sin \psi) \quad (2)$$

$$\dot{z} = -u \sin \theta + v \cos \theta \sin \phi + w \cos \theta \cos \phi \quad (3)$$

where s and c correspond to the \sin and \cos of the angles. In similar manner, the body-to-earth transformation can also be applied to the angular rates, yielding the following equations:

$$\dot{\phi} = p + r \cos \phi \tan \theta + q \sin \phi \tan \theta \quad (4)$$

$$\dot{\theta} = q \cos \phi - r \sin \phi \quad (5)$$

$$\dot{\psi} = (r \cos \phi + q \sin \phi) / \cos \theta \quad (6)$$

The force equation in translational motion in body-fixed frame is: $\vec{F}_{\text{ext}} = m (\dot{\vec{v}}_B + \vec{\Omega}_B \times \vec{v}_B)$. In similar manner, the moment equation in rotational motion in body-fixed frame is: $\vec{M}_{\text{ext}} = I_B \dot{\vec{\Omega}}_B + \vec{\Omega}_B \times (I_B \vec{\Omega}_B)$. The force \vec{F}_{ext} and the moment \vec{M}_{ext} are the resultant external force and moment from various sources: aerodynamics, propulsion, and gravity.

Expanding the force and moment equations in three translational and three rotational axes yield:

$$F_{\text{ext},x} = m (\dot{u} + q w - r v) \quad (7)$$

$$F_{\text{ext},y} = m (\dot{v} - p w + r u) \quad (8)$$

$$F_{\text{ext},z} = m (\dot{w} + p v - q u) \quad (9)$$

$$M_{\text{ext},x} = I_{xx} \dot{p} - I_{xz} \dot{r} - q (I_{zx} p - I_{zz} r) - I_{yy} q r \quad (10)$$

$$M_{\text{ext},y} = I_{yy} \dot{q} + p (I_{zx} p - I_{zz} r) + r (I_{xx} p - I_{xz} r) \quad (11)$$

$$M_{\text{ext},z} = I_{zz} \dot{r} - I_{zx} \dot{p} - q (I_{xx} p - I_{xz} r) + I_{yy} p q \quad (12)$$

where m is the mass of the aircraft, I is the aircraft moment of inertia. Equations (??-??) are used to solve for the translational acceleration in body-fixed frame: $\dot{\vec{v}}_B = (\dot{u}, \dot{v}, \dot{w})^T$, while Equations (??-??) are used to solve for the rotational acceleration in body-fixed frame: $\dot{\vec{\Omega}}_B = (\dot{p}, \dot{q}, \dot{r})^T$.

Finally, Equations (??-??) represent the non-linear, coupled, first order system of equations that describe the 6-DOF aircraft motion. The details of the derivation for each of the external forces \vec{F}_{ext} and moments \vec{M}_{ext} can be found in Appendix ?? for the aerodynamics and Appendix ?? for propulsion. The atmosphere is modeled using the International Standard Atmosphere (ISA) [?], see Appendix ?? for the details.

III. AIRCRAFT PITCH DYNAMIC MODELING

A. Trim condition

Let's define a vector of motion states as:

$$\mathbf{x} = (x, y, z, \phi, \theta, \psi, p, q, r, u, v, w)^T \quad (13)$$

a vector of outputs as:

$$\mathbf{y} = (\mathbf{x}, \alpha, \beta, \gamma, \zeta, v_{\text{TAS}}, \rho)^T \quad (14)$$

and a vector of control inputs as:

$$\mathbf{u} = (\delta_e, \delta_a, \delta_r, \delta_t)^T \quad (15)$$

where δ_e , δ_a , and δ_r are the deflection angles of elevator, aileron, and rudder, respectively, while δ_t is the throttle input. Assume the aircraft is flying a level unaccelerated cruise at 5000 ft (1524 m). The trim condition solves:

$$\mathbf{x}_0 = (x_0, y_0, z_0, \phi_0, \theta_0, \psi_0, p_0, q_0, r_0, u_0, v_0, w_0)^T \quad (16)$$

$$\dot{\mathbf{x}}_0 = f(\mathbf{x}_0) = 0 \quad (17)$$

By assuming all the values for Cessna 172 aircraft [?], the above equations are solved in Matlab. The following solutions are found for the trimmed states, control inputs, and outputs.

$$\mathbf{x}_0 = (0, 0, -1524, 0, 0, 0, 62.3866, 0, 0, 0, 0, 0)^T \quad (18)$$

$$\mathbf{u}_0 = (-0.0032115, 0, 0, 0.6792)^T \quad (19)$$

$$\mathbf{y}_0 = (\mathbf{x}_0, 0, 0, 0, 0, 62.3866, 1.0557)^T \quad (20)$$

B. Linearized equations

Equations (??-??) are non-linear and coupled. So, they need to be linearized. Linearization procedure assumes a small deviation from the previously trimmed states, i.e., $\mathbf{x} = \mathbf{x}_0 + \Delta \mathbf{x}$. Therefore, the control input will be an additional input from the trimmed inputs, i.e., $\mathbf{u} = \mathbf{u}_0 + \Delta \mathbf{u}$. In similar manner, the outputs will be $\mathbf{y} = \mathbf{y}_0 + \Delta \mathbf{y}$. The linearization procedure is performed in Matlab, again by assuming the values for Cessna 172 aircraft, the following linear equations are obtained.

$$\Delta \dot{\mathbf{x}} = A \Delta \mathbf{x} + B \Delta \mathbf{u} \quad (21)$$

$$\Delta \mathbf{y} = C \Delta \mathbf{x} + D \Delta \mathbf{u} \quad (22)$$

where matrices A , B , C , and D are the state space representation of the linearized twelve states differential equations. The numerical values details are presented in Appendix ?? and ??.

Matrices A , B , C , and D can be separated into two uncoupled equations: longitudinal and lateral. Since we focus on pitch control, only longitudinal equations are used to design our controller. In longitudinal motion, the states are reduced into six states, the control inputs are reduced into two, and the outputs into ten:

$$\Delta \mathbf{x}_{\text{long}} = (\Delta x, \Delta z, \Delta \theta, \Delta u, \Delta w, \Delta q)^T \quad (23)$$

$$\Delta \mathbf{u}_{\text{long}} = (\Delta \delta_e, \Delta \delta_t)^T \quad (24)$$

$$\Delta \mathbf{y}_{\text{long}} = (\Delta \mathbf{x}_{\text{long}}, \Delta \alpha, \Delta \gamma, \Delta v_{\text{TAS}}, \Delta \rho)^T \quad (25)$$

The resulting state-space representation for longitudinal motion is then written:

$$\Delta \dot{\mathbf{x}}_{\text{long}} = A_{\text{long}} \Delta \mathbf{x} + B_{\text{long}} \Delta \mathbf{u} \quad (26)$$

$$\Delta \mathbf{y}_{\text{long}} = C_{\text{long}} \Delta \mathbf{x} + D_{\text{long}} \Delta \mathbf{u} \quad (27)$$

$$A_{\text{long}} = \begin{pmatrix} 0 & 0 & 0 & 1.0 & 0 & 0 \\ 0 & 0 & -62.39 & 0 & 1.0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.0 \\ 0 & -0.0001 & -9.807 & -0.0477 & 0.2388 & 0 \\ 0 & -0.0022 & 0 & -0.3152 & -2.64 & 60.9 \\ 0 & 0 & 0 & 0.0005 & -0.2494 & -3.971 \end{pmatrix} \quad (28)$$

$$B_{\text{long}} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1.91 & 1.462 \\ -13.69 & 0.0255 \\ -33.99 & -0.0146 \end{pmatrix} \quad (29)$$

$$C_{\text{long}} = \begin{pmatrix} 1.0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.0 \\ 0 & 0 & 0 & 0 & 0.016 & 0 \\ 0 & 0 & 1.0 & 0 & -0.016 & 0 \\ 0 & 0 & 0 & 1.0 & 0 & 0 \\ 0 & 0.0002 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (30)$$

$$D_{\text{long}} = \mathbf{0}^{10 \times 3} \quad (31)$$

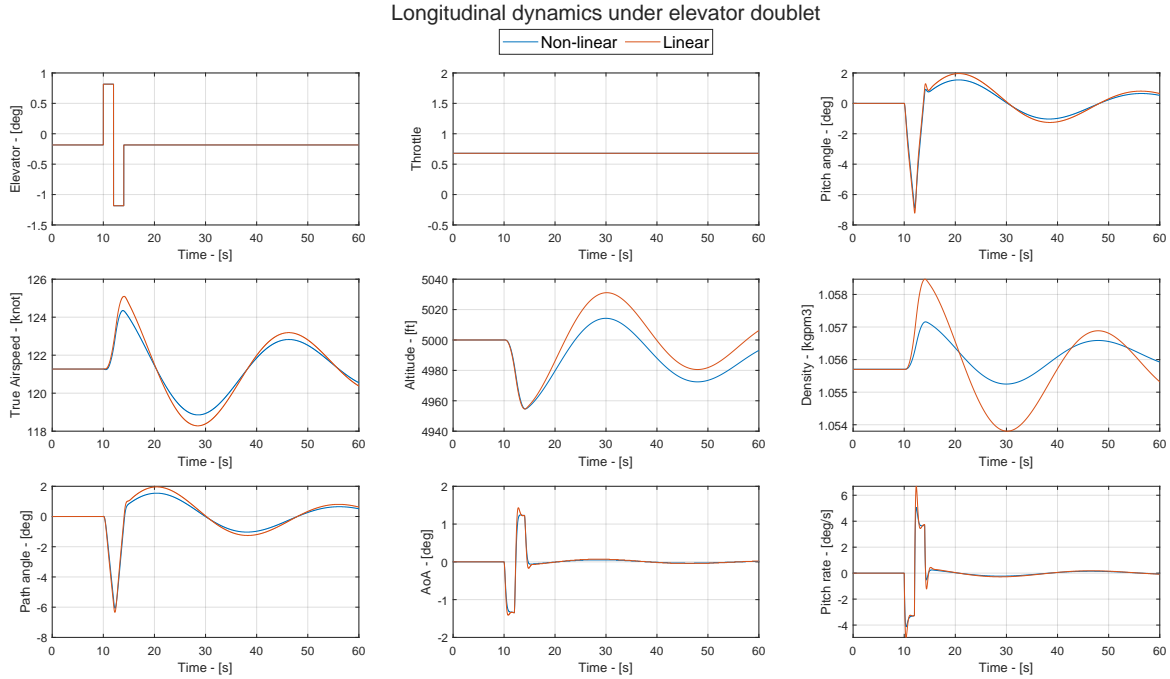


Fig. 1: Longitudinal dynamics under elevator doublet: nonlinear vs linear models.

C. Verification of linear model: elevator doublet

The linear models are compared against the non-linear model under an elevator doublet. Figure ?? shows the results of ± 1 degree elevator doublet using the linear and non-linear models. From the pitch angle dynamics, it is safe to say that our linear model agrees with the non-linear model. It is noted that the air density also varies as the altitude varies. In the same way, the true airspeed also changes as the pitch changes. These are often times neglected by other works [?], [?], [?], [?].

D. Open-loop and closed-loop response

The open-loop and closed-loop (with unity feedback) systems are checked against a step response of 0.2 rad ($\approx 11.46 \text{ deg}$). Figure ?? and ?? show the open-loop and closed-loop step responses, respectively. The open-loop response is marginally stable, but difficult to control and the steady-state error is also large. The unity feedback even makes the system unstable.

IV. TRADITIONAL METHOD

A. Static PID controller

Proportional-Integral-Derivative (PID) control is a well-established control method that has been used across engineering fields due to its simplicity. By tuning the proper gain values, the performance of a closed-loop system can be enhanced. The mathematical formulation of the PID in this project is written as:

$$C_{\text{PID}}(s) = K_P + \frac{K_I}{s} + \frac{K_D s}{s/N + 1} \quad (32)$$

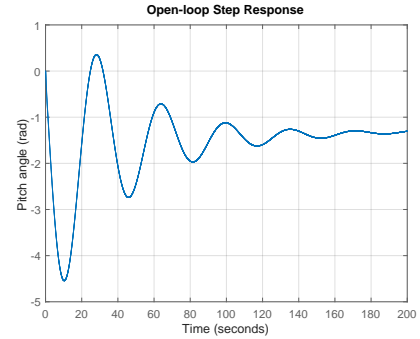


Fig. 2: Step response of the open-loop system.

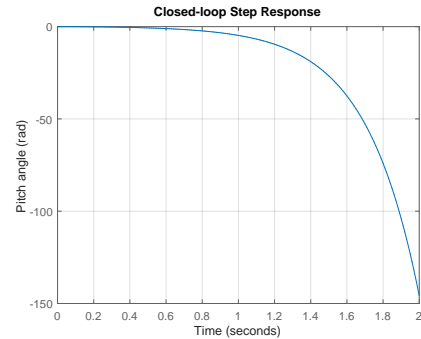


Fig. 3: Step response of the closed-loop with unity feedback.

where K_P is the proportional gain, K_I is the integral gain, K_D is the derivative gain, and N is the time filter coefficient taken as a constant 100 in this project.

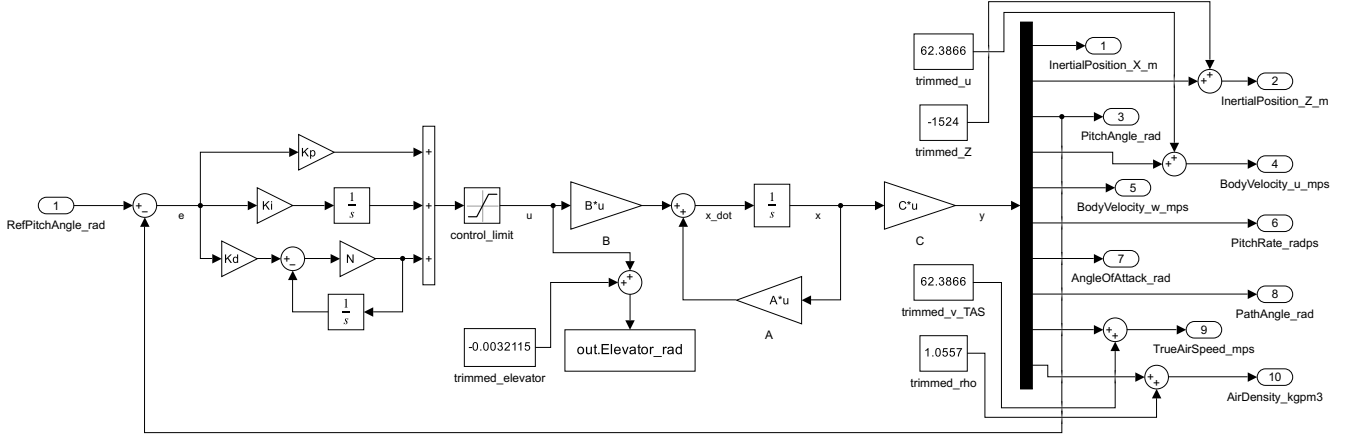


Fig. 4: Static PID control coupled with longitudinal dynamic system: tracking the reference pitch angle.

TABLE I: Time-domain performance of PID with different static gains

PID Name	K_P	K_I	K_D	Rise Time (s)	Settling Time (s)	Overshoot (%)	Steady-State Error (%)
PID_1	-1.0	-1.0	0.0	0.2370	3.1187	22.4851	0.5179
PID_2	-1.0	-0.8	0.0	0.2429	3.5128	19.0088	0.6609
PID_3	-1.0	-0.6	0.0	0.2488	4.0294	15.6260	0.8921
PID_4	-1.0	-0.3	0.0	0.2648	5.0701	9.9522	1.4383
PID_5	-1.0	-0.3	-0.1	0.2848	4.9752	4.3979	1.1606

The schematic of the static PID coupled with aircraft longitudinal dynamics is shown in Figure ?? . It is noted that although the system has a multiple outputs, the output being tracked is only the pitch angle. Therefore, the system is essentially still a Single-Input-Single-Output (SISO) model. The input from the throttle is assumed to be constant at the trimmed value, i.e. $\Delta\delta_t = 0$ or $\delta_{t_0} = 0.6792$. It is also noted that the gains K_P , K_I , and K_D are constant and tuned beforehand.

B. Time-domain performance

A controller should be designed to have the following criteria under a 0.2rad step input: a) rise time < 2 seconds, b) settling time < 10 seconds, c) overshoot $< 10\%$, and d) steady-state error $< 2\%$. PID controller with the static gains listed in Table ?? are explored. In PID_1, the derivative gain is zero. The rise and settling times match the criteria, while the overshoot of 22.4851% does not meet the $< 10\%$ criterion. Thus, the integral gain is iteratively reduced until it reaches $K_I = -0.3$ in PID_4, where it satisfies all our criteria. But, when the derivative gain is introduced in PID_5, the system's overshoot is reduced to 4.3979%. It is noted that all the systems have steady-state error $< 2\%$.

C. Tracking performance and input limitation

Figure ?? shows the performance of PID with different static gains tracking the 0.2 rad step reference input. As discussed previously, all the static PID controllers successfully track the reference input. However, this plot does not give any information regarding the requested control input by each controller. Therefore, the input-output dynamics are plotted in

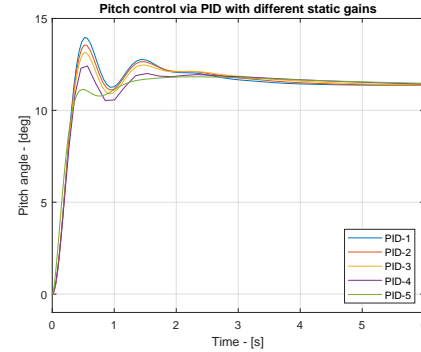


Fig. 5: Tracking performance of PID with different gains.

Figure ?? . It is observed that PID_5 requests a high control input from the very beginning: a maximum of -30° of elevator deflection. It is noted that our closed-loop system has a saturation model to prevent the controller to give a very high control input, i.e., $-30^\circ \leq \Delta\delta_e \leq 30^\circ$. If the saturation is removed, PID_5 will request around -100° , which is outside the capability of the mechanical system.

V. INTELLIGENT METHOD

A. RL-based adaptive PID controller

The class of RL algorithm used here is called PPO [?], in which it includes two deep neural networks: action net and value net. Specifically, the action net is trained to map a normalized error of the aircraft pitch (difference between the desired and observed pitch divided by the desired pitch), as an observation, into the three PID gains as its actions.

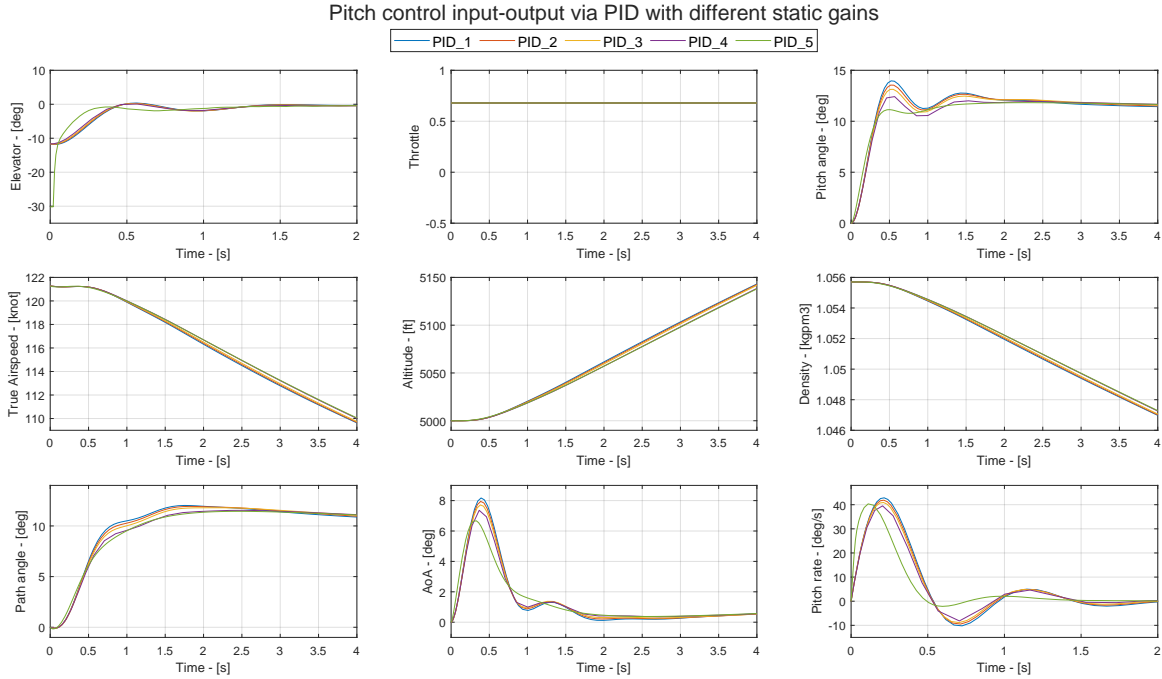


Fig. 6: Input-output dynamics of the systems with different static PID gains.

Simultaneously, the value net is trained to map the given observation and actions into a reward value.

B. Training and validation of different neural net architectures

The key idea of any RL algorithms is to train an agent (action net) by letting it interact with the environment with the purpose of cumulating the highest reward possible. After training, the agent can be deployed to perform the specific task it was trained for. In this project, the agent interacts with the aircraft longitudinal system + PID controller as its environment by giving actions (PID gains) with the purpose of obtaining the highest reward. The environment starts with $\theta_{\text{obs}} = 0^\circ$, and the desired pitch is $-0.5 \leq \theta_{\text{des}} \leq 0.5$, measured in *radians*. The reward function is defined as:

$$\text{rew} = - \left(\frac{\theta_{\text{des}} - \theta_{\text{obs}}}{\theta_{\text{des}}} \right)^2 - \left(\frac{\Delta \delta_e}{\Delta \delta_{e,\text{max}}} \right)^2 - 10.0 \delta_{\text{term}} + 1.0 \quad (33)$$

The first term corresponds to the punishment for deviating from the desired pitch value, the second term dictates the agent to act efficiently, the third term is the punishment if the $|\theta_{\text{obs}}| > 90^\circ$ ($\delta_{\text{term}} = 1$), while the last term is the reward for advancing further in timesteps. The maximum time and the discrete timestep are set to 6s and 0.01s, respectively.

Unlike traditional supervised learning algorithms, RL (specifically PPO), collects data and trains its networks simultaneously, which is referred to as an on-policy algorithm. During the training, a separate set of environments are used as validation set to evaluate how well the agent performs the task considered (pitch control). The training is stopped when the mean reward collected by the agent in the validation set reach a certain value. In this project, this value is set to 580,

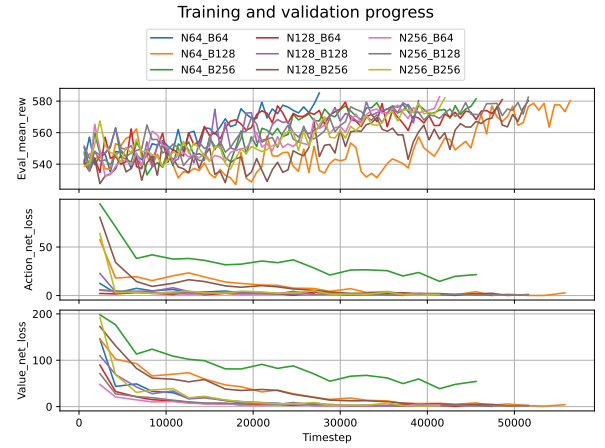


Fig. 7: Training and validation progress.

having known that the maximum reward is 600 per episode. This way, the training will not be too exhaustive, while the trained agent is good enough at performing its task.

Different architectures for both the action and value nets are investigated by adjusting the number of neurons (N) in each hidden layer of the nets to be either 64, 128, or 256. The batchsize (B) is also varied to be either 64, 128, or 256. The number of hidden layers is kept at 2 with *tanh* activation function. The action net output is clipped between $[-1, 1]$ and rescaled to be between $[-3, 0]$ to produce the PID gains.

Figure ?? shows how the training progresses for each architecture. It is observed that the training losses for both action and value nets are decreasing, while all the agents successfully reach the mean reward threshold in the validation

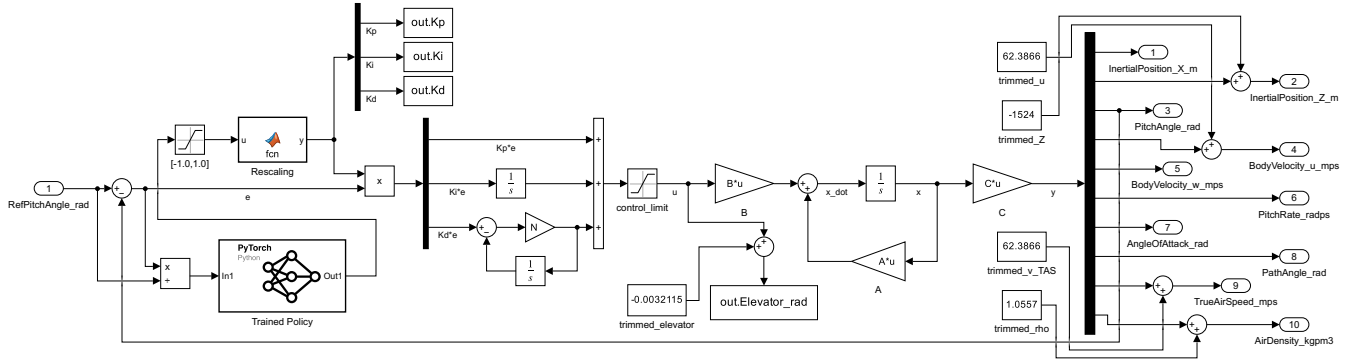


Fig. 8: RL-based adaptive PID controller being deployed for tracking the aircraft pitch angle.

TABLE II: Time-domain performance of RL-based adaptive PID with different neural network architectures

No	N_neurons	Batchsize	Req. Training Timesteps	Rise Time (s)	Settling Time (s)	Overshoot (%)	Steady-State Error (%)
1	64	64	27600	0.1220	3.7462	4.2309	0.4396
2	64	128	56400	0.1543	5.1725	10.6501	0.9282
3	64	256	45600	0.1606	3.3941	8.3370	0.3259
4	128	64	48600	0.4991	4.7149	9.7276	0.5768
5	128	128	39000	0.1371	3.9755	4.9228	0.4647
6	128	256	51600	0.4823	3.9777	4.5844	0.4058
7	256	64	41400	0.1269	3.8706	7.6882	0.2827
8	256	128	51600	0.1313	4.4782	9.4892	0.6104
9	256	256	42000	0.1280	5.8598	7.0842	0.6672

sets. The required training timesteps is also summarized in Table ???. The less the timesteps, the faster the training is, i.e., the fastest to train is the N64_B64, while the slowest is the N64_B128.

The training is performed using stable-baselines3 [?]. Subsequently, the trained policy (action net) is saved as a pytorch model [?], which can be called in Simulink as a toolbox to predict the adaptive PID gains, shown in Figure ?. It is noted that the PID gains are changing depending on the observation (normalized pitch error).

C. Time-domain performance: testing

After a trained policy is obtained and deployed in the Simulink model (see Figure ??), the testing is performed under a 0.2rad step input, as previously done with the static PID. The controllers' step-response characteristics are summarized in Table ??, with the same criteria as before. It can be observed that all trained policies with different net architectures, except the one with 64 neurons and 128 batchsize (N64_B128), satisfy all the criteria with the rise time $< 0.5\text{s}$, the settling time $< 6\text{s}$, the overshoot $< 10\%$, and the steady-state error $< 1\%$. The N64_B128 policy fails to satisfy the $< 10\%$ overshoot criterion by a small amount of violation.

D. Tracking performance and adaptive PID gains trajectories

Based on the previous results, three trained policies are selected for comparison. The pitch trajectories by these selected trained policies are depicted in Figure ??, while Figure ?? illustrates a more comprehensive input-output dynamics. It is observed that the trained policies request a maximum control input $|\Delta\delta_e| = 30^\circ$ at $t < 0.5\text{s}$, resulting in short rise

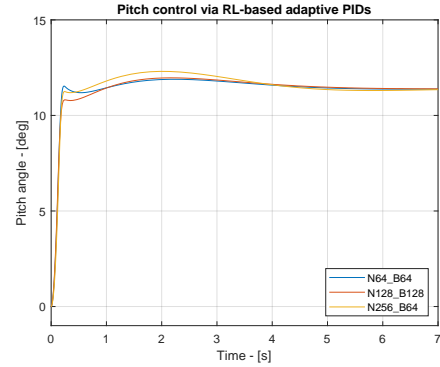


Fig. 9: Tracking performance of selected RL-based PIDs.

times. In Figure ??, the trajectories of PID gains over time are demonstrated. Each set of PID gains displays a transient behavior $t < 0.5\text{s}$ before converging to a near-steady state towards the end. These varying PID gains contribute to the great time-domain performances.

VI. CONCLUSION AND FUTURE WORK

In this project, a 6-DOF non-linear aircraft dynamics model based on Cessna-172 stability derivatives is developed from first principles. Subsequently, a linearized system is derived and used to design PID controllers for tracking a desired pitch angle. Two methods are used: traditional and intelligent. In the traditional method, manual tuning of static PID gains is conducted, while the intelligent method uses an RL algorithm to predict adaptive PID gains. The former relies on a trial-and-error process assuming familiarity with the dynamic model, whereas the latter involves crafting an appropriate reward func-

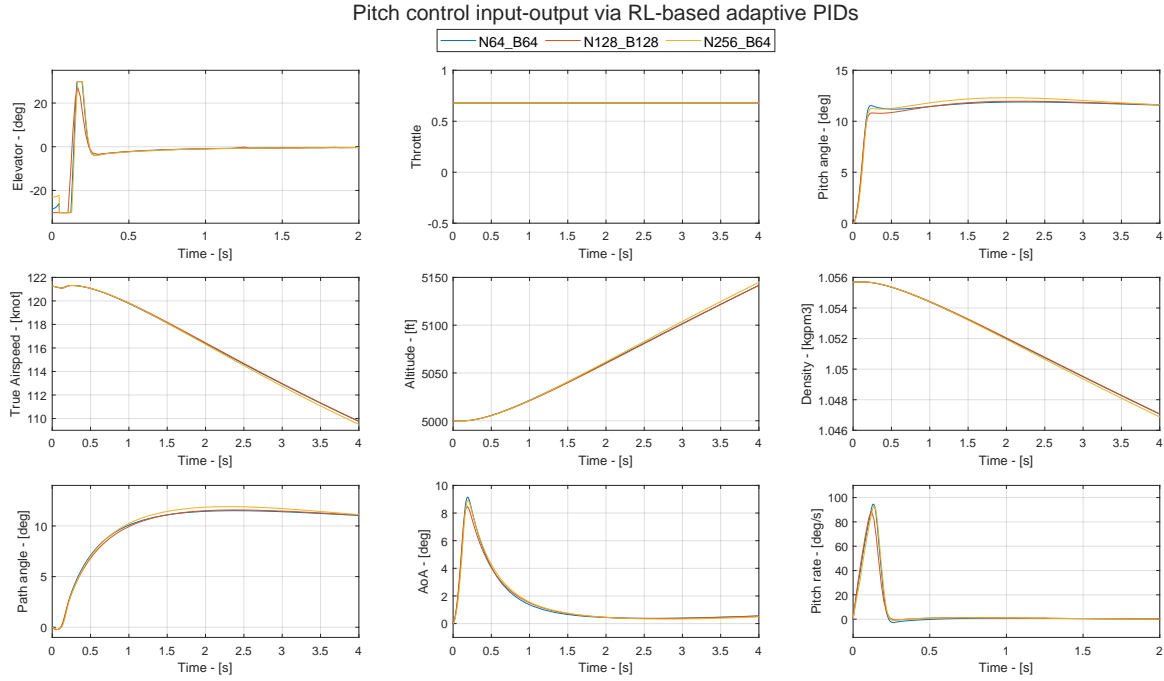


Fig. 10: Input-output dynamics of the systems with RL-based adaptive PIDs.

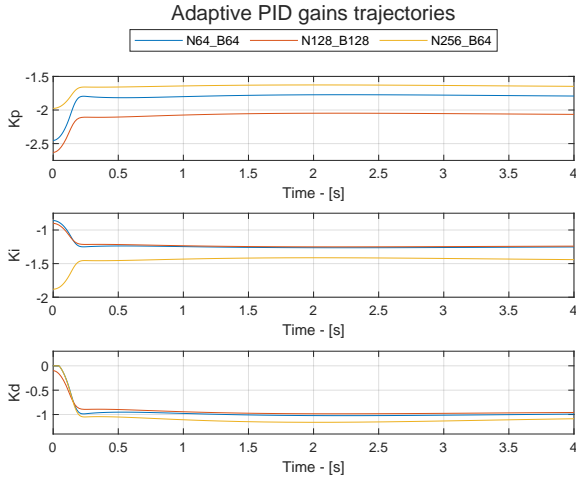


Fig. 11: Adaptive PID gains for selected trained policies.

tion and network architectures without requiring knowledge about the dynamic model (model-free, data-driven). Future work should investigate the effectiveness of adaptive PID gains in scenarios involving time-varying uncertainties, e.g. wind gusts. Additionally, it should assess off-design situations where static gains could potentially exhibit degraded performance.

REFERENCES

- [1] R.C. Nelson, "Flight Stability and Automatic Control," McGraw Hill, Second Edition, 1998.
- [2] Bernard Etkin, and Lloyd Duff Reid, "Dynamics of Flight, Stability and Control," McGrawHill, Third Edition, 1996.
- [3] Duan, M., Cesnik, C. E. S., Kolmanovsky, I. V., and Vetrano, F., "Control-Oriented Modeling for Flexible Aircraft," *Journal of Aircraft*, Vol. 61, No. 1, 2024, pp. 183–195. <https://doi.org/10.2514/1.C037049>
- [4] S. N. Deepa and G. Sudha, "Longitudinal control of aircraft dynamics based on optimization of PID parameters," *Thermophys. Aeromech.*, vol. 23, no. 2, pp. 185–194, 2016, doi: 10.1134/S0869864316020049.
- [5] Vishal and J. Ohri, "GA tuned LQR and PID controller for aircraft pitch control," presented at the *2014 IEEE 6th India International Conference on Power Electronics (IICPE)*, Kurukshestra, India: IEEE, 2014, pp. 1–6. doi: 10.1109/IICPE.2014.7115839.
- [6] H. Arrosida and M. E. Echsony, "Aircraft Pitch Control Design using Observer-State Feedback Control," *KINETIK*, pp. 263–272, Sep. 2017, doi: 10.22219/kinetik.v2i4.267.
- [7] A. Mohanty, and E. Schneider, "Tuning of an Aircraft Pitch PID Controller with Reinforcement Learning and Deep Neural Net," (2019). Accessed: Dec. 12, 2024. [Online]. Available: https://cs229.stanford.edu/proj2019aut/data/assignment_308832_raw/26643693.pdf
- [8] N. Wahid and N. Hassan, "Self-Tuning Fuzzy PID Controller Design for Aircraft Pitch Control," presented at the *2012 3rd International Conference on Intelligent Systems, Modelling and Simulation (ISMS)*, Kota Kinabalu, Malaysia: IEEE, 2012, pp. 19–24. doi: 10.1109/ISMS.2012.27.
- [9] Y. Liang, G. Li, M. Xu, J. Zhao, F. Hao, and H. Shi, "An intelligent control method based on artificial neural network for numerical flight simulation of the basic finner projectile with pitching maneuver," *Defence Technology*, vol. 32, pp. 663–674, 2024, doi: 10.1016/j.dt.2023.07.012.
- [10] D. J. Richter, R. A. Calix, and K. Kim, "A Review of Reinforcement Learning for Fixed-Wing Aircraft Control Tasks," *IEEE Access*, vol. 12, pp. 103026–103048, 2024, doi: 10.1109/ACCESS.2024.3433540.
- [11] T. P. Lillicrap et al., "Continuous control with deep reinforcement learning," 2015, arXiv. doi: 10.48550/ARXIV.1509.02971.
- [12] C. Kasnakoğlu, "Investigation of Multi-Input Multi-Output Robust Control Methods to Handle Parametric Uncertainties in Autopilot Design," *PLoS ONE*, vol. 11, no. 10, p. e0165017, Oct. 2016, doi: 10.1371/journal.pone.0165017.

- [13] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal Policy Optimization Algorithms,” 2017, arXiv. doi: 10.48550/ARXIV.1707.06347.
- [14] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, “Stable-Baselines3: Reliable Reinforcement Learning Implementations,” *Journal of Machine Learning Research*, vol. 22, no. 268, pp. 1–8, 2021, Accessed: Dec. 12, 2024. [Online]. Available: <http://jmlr.org/papers/v22/20-1364.html>
- [15] A. Paszke et al., “PyTorch: An Imperative Style, High-Performance Deep Learning Library,” 2019, arXiv. doi: 10.48550/ARXIV.1912.01703.
- [16] “U.S. Standard Atmosphere,” U.S. Government Printing Office, Washington, D.C., 1976.

APPENDIX A CESSNA-172 DATA [?]

Wing Data

$$\text{wing area } S = 16.1651 \text{ m} \quad (34)$$

$$\text{wingspan } b = 10.9118 \text{ m} \quad (35)$$

$$\text{MAC } c = 1.4935 \text{ m} \quad (36)$$

Mass Properties Data

$$\text{Mass } m = 1043.3 \text{ kg} \quad (37)$$

$$\text{CG}_{\text{mac}} = 0.3 \quad (38)$$

$$I_{XX} = 1285.3 \text{ kgm}^2 \quad (39)$$

$$I_{YY} = 1824.9 \text{ kgm}^2 \quad (40)$$

$$I_{ZZ} = 2666.9 \text{ kgm}^2 \quad (41)$$

$$I_{XZ} = 0.0 \text{ kgm}^2 \quad (42)$$

$$y_{\text{CG}} = 0.0 \text{ m} \quad (43)$$

$$z_{\text{CG}} = 0.2 \text{ m} \quad (44)$$

$$g = 9.80665 \text{ m/s}^2 \quad (45)$$

Engine Data

$$v_{\text{ref}} = 51.4 \text{ m/s} \quad (46)$$

$$\rho_{\text{ref}} = 1.225 \text{ kg/m}^3 \quad (47)$$

$$n_v = -1.0 \quad (48)$$

$$n_\rho = 0.75 \quad (49)$$

$$\alpha_F = 1.0^\circ \quad (50)$$

$$X_F = 1.0 \text{ m} \quad (51)$$

$$Z_F = 0.0 \text{ m} \quad (52)$$

$$T_{\text{max}} = 2070.0 \text{ N} \quad (53)$$

Aerodynamic Derivative: Lift Coefficient

$$C_{L_0} = 0.31 \quad (54)$$

$$C_{L_\alpha} = 5.143 \text{ /rad} \quad (55)$$

$$C_{L_{\delta_e}} = 0.43 \text{ /rad} \quad (56)$$

$$C_{L_{\dot{\alpha}}} = 0.0 \text{ s/rad} \quad (57)$$

$$C_{L_q} = 3.9 \text{ s/rad} \quad (58)$$

Aerodynamic Derivative: Drag Coefficient

$$C_{D_0} = 0.031 \quad (59)$$

$$C_{D_\alpha} = 0.13 \text{ /rad} \quad (60)$$

$$C_{D_{\delta_e}} = 0.06 \text{ /rad} \quad (61)$$

Aerodynamic Derivative: Pitching Moment Coefficient

$$C_{m_0} = -0.015 \quad (62)$$

$$C_{m_\alpha} = -0.89 \text{ /rad} \quad (63)$$

$$C_{m_{\delta_e}} = -1.28 \text{ /rad} \quad (64)$$

$$C_{m_{\dot{\alpha}}} = -7.27 \text{ s/rad} \quad (65)$$

$$C_{m_q} = -12.4 \text{ s/rad} \quad (66)$$

Aerodynamic Derivative: Side Force Coefficient

$$C_{Y_\beta} = -0.31 \text{ /rad} \quad (67)$$

$$C_{Y_{\delta_a}} = 0.0 \text{ /rad} \quad (68)$$

$$C_{Y_{\delta_r}} = 0.187 \text{ /rad} \quad (69)$$

$$C_{Y_p} = -0.037 \text{ s/rad} \quad (70)$$

$$C_{Y_r} = 0.21 \text{ s/rad} \quad (71)$$

Aerodynamic Derivative: Rolling Moment Coefficient

$$C_{l_\beta} = -0.089 \text{ /rad} \quad (72)$$

$$C_{l_{\delta_a}} = -0.178 \text{ /rad} \quad (73)$$

$$C_{l_{\delta_r}} = 0.0147 \text{ /rad} \quad (74)$$

$$C_{l_p} = -0.47 \text{ s/rad} \quad (75)$$

$$C_{l_r} = 0.096 \text{ s/rad} \quad (76)$$

Aerodynamic Derivative: Yawing Moment Coefficient

$$C_{n_\beta} = 0.065 \text{ /rad} \quad (77)$$

$$C_{n_{\delta_a}} = -0.053 \text{ /rad} \quad (78)$$

$$C_{n_{\delta_r}} = -0.0657 \text{ /rad} \quad (79)$$

$$C_{n_p} = -0.03 \text{ s/rad} \quad (80)$$

$$C_{n_r} = -0.099 \text{ s/rad} \quad (81)$$

APPENDIX B
NON-LINEAR COUPLED 6-DOF SYSTEM OF EQUATIONS

$$\mathcal{L}_\phi = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{pmatrix} \quad (82)$$

$$\mathcal{L}_\theta = \begin{pmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{pmatrix} \quad (83)$$

$$\mathcal{L}_\psi = \begin{pmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (84)$$

$$\mathcal{L}_{E \rightarrow B} = \mathcal{L}_\phi \mathcal{L}_\theta \mathcal{L}_\psi \quad (85)$$

$$\mathcal{L}_{B \rightarrow E} = \mathcal{L}_\psi^{-1} \mathcal{L}_\theta^{-1} \mathcal{L}_\phi^{-1} \quad (86)$$

$$\vec{v}_E = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} = \mathcal{L}_{B \rightarrow E} \vec{v}_B = \mathcal{L}_{B \rightarrow E} \begin{pmatrix} u \\ v \\ w \end{pmatrix} \quad (87)$$

$$\dot{x} = w (\sin(\phi) \sin(\psi) + \cos(\phi) \cos(\psi) \sin(\theta)) - v (\cos(\phi) \sin(\psi) - \cos(\psi) \sin(\phi) \sin(\theta)) + u \cos(\psi) \cos(\theta) \quad (88)$$

$$\dot{y} = v (\cos(\phi) \cos(\psi) + \sin(\phi) \sin(\psi) \sin(\theta)) - w (\cos(\psi) \sin(\phi) - \cos(\phi) \sin(\psi) \sin(\theta)) + u \cos(\theta) \sin(\psi) \quad (89)$$

$$\dot{z} = w \cos(\phi) \cos(\theta) - u \sin(\theta) + v \cos(\theta) \sin(\phi) \quad (90)$$

$$\vec{W}_B = \mathcal{L}_{E \rightarrow B} \vec{W}_E = \mathcal{L}_{E \rightarrow B} \begin{pmatrix} 0 \\ 0 \\ m g \end{pmatrix} \quad (91)$$

$$\vec{W}_B = \begin{pmatrix} -m g \sin(\theta) \\ m g \cos(\theta) \sin(\phi) \\ m g \cos(\phi) \cos(\theta) \end{pmatrix} \quad (92)$$

$$\vec{F}_{\text{ext}} = \vec{F}_{\text{aero}} + \vec{F}_{\text{thrust}} + \vec{W}_B \quad (93)$$

$$\vec{F}_{\text{ext}} = \begin{pmatrix} F_{\text{ext},x} \\ F_{\text{ext},y} \\ F_{\text{ext},z} \end{pmatrix} = \begin{pmatrix} F_{\text{aero},x} + F_{\text{thrust},x} - m g \sin(\theta) \\ F_{\text{aero},y} + F_{\text{thrust},y} + m g \cos(\theta) \sin(\phi) \\ F_{\text{aero},z} + F_{\text{thrust},z} + m g \cos(\phi) \cos(\theta) \end{pmatrix} \quad (94)$$

$$\vec{M}_{\text{ext}} = \vec{M}_{\text{aero}} + \vec{M}_{\text{thrust}} \quad (95)$$

$$\vec{M}_{\text{ext}} = \begin{pmatrix} M_{\text{ext},x} \\ M_{\text{ext},y} \\ M_{\text{ext},z} \end{pmatrix} = \begin{pmatrix} M_{\text{aero},x} + M_{\text{thrust},x} \\ M_{\text{aero},y} + M_{\text{thrust},y} \\ M_{\text{aero},z} + M_{\text{thrust},z} \end{pmatrix} \quad (96)$$

$$\vec{\Omega}_B = \begin{pmatrix} p \\ q \\ r \end{pmatrix} = \begin{pmatrix} \dot{\phi} - \dot{\psi} \sin(\theta) \\ \dot{\theta} \cos(\phi) + \dot{\psi} \cos(\theta) \sin(\phi) \\ \dot{\psi} \cos(\phi) \cos(\theta) - \dot{\theta} \sin(\phi) \end{pmatrix} \quad (97)$$

$$\dot{\phi} = p + r \cos(\phi) \tan(\theta) + q \sin(\phi) \tan(\theta) \quad (98)$$

$$\dot{\theta} = q \cos(\phi) - r \sin(\phi) \quad (99)$$

$$\dot{\psi} = \frac{r \cos(\phi) + q \sin(\phi)}{\cos(\theta)} \quad (100)$$

$$\vec{F}_{\text{ext}} = m \left(\dot{\vec{V}}_{\text{B}} + \vec{\Omega}_{\text{B}} \times \vec{V}_{\text{B}} \right) \quad (101)$$

$$\vec{F}_{\text{ext}} = m \left(\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} + \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} \times \begin{bmatrix} u \\ v \\ w \end{bmatrix} \right) = m \begin{pmatrix} \dot{u} + q w - r v \\ \dot{v} - p w + r u \\ \dot{w} + p v - q u \end{pmatrix} \quad (102)$$

$$F_{\text{aero},x} + F_{\text{thrust},x} - m g \sin(\theta) = m (\dot{u} + q w - r v) \quad (103)$$

$$F_{\text{aero},y} + F_{\text{thrust},y} + m g \cos(\theta) \sin(\phi) = m (\dot{v} - p w + r u) \quad (104)$$

$$F_{\text{aero},z} + F_{\text{thrust},z} + m g \cos(\phi) \cos(\theta) = m (\dot{w} + p v - q u) \quad (105)$$

$$\vec{M}_{\text{ext}} = I_{\text{B}} \dot{\vec{\Omega}}_{\text{B}} + \vec{\Omega}_{\text{B}} \times (I_{\text{B}} \vec{\Omega}_{\text{B}}) \quad (106)$$

$$\vec{M}_{\text{ext}} = \begin{bmatrix} I_{xx} & 0 & -I_{xz} \\ 0 & I_{yy} & 0 \\ -I_{zx} & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} + \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \left(\begin{bmatrix} I_{xx} & 0 & -I_{xz} \\ 0 & I_{yy} & 0 \\ -I_{zx} & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \right) \quad (107)$$

$$M_{\text{aero},x} + M_{\text{thrust},x} = I_{xx} \dot{p} - I_{xz} \dot{r} - q (I_{zx} p - I_{zz} r) - I_{yy} q r \quad (108)$$

$$M_{\text{aero},y} + M_{\text{thrust},y} = I_{yy} \dot{q} + p (I_{zx} p - I_{zz} r) + r (I_{xx} p - I_{xz} r) \quad (109)$$

$$M_{\text{aero},z} + M_{\text{thrust},z} = I_{zz} \dot{r} - I_{zx} \dot{p} - q (I_{xx} p - I_{xz} r) + I_{yy} p q \quad (110)$$

$$\mathbf{x} = (x, y, z, \phi, \theta, \psi, p, q, r, u, v, w)^T \quad (111)$$

$$\dot{\mathbf{x}} = f(\mathbf{x}) = f(x, y, z, \phi, \theta, \psi, p, q, r, u, v, w) \quad (112)$$

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \\ \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{pmatrix} = \begin{pmatrix} u C_{\theta} C_{\psi} + v (S_{\phi} S_{\theta} C_{\psi} - C_{\phi} S_{\psi}) + w (C_{\phi} S_{\theta} C_{\psi} + S_{\phi} S_{\psi}) \\ u C_{\theta} S_{\psi} + v (S_{\phi} S_{\theta} S_{\psi} + C_{\phi} C_{\psi}) + w (C_{\phi} S_{\theta} S_{\psi} - S_{\phi} C_{\psi}) \\ -u S_{\theta} + v S_{\phi} C_{\theta} + w C_{\phi} C_{\theta} \\ p + q S_{\phi} T_{\theta} + r C_{\phi} T_{\theta} \\ q C_{\phi} - r S_{\phi} \\ (q S_{\phi} + r C_{\phi}) / C_{\theta} \\ r v - q w + F_{\text{ext},x} / m \\ p w - r u + F_{\text{ext},y} / m \\ q u - p v + F_{\text{ext},z} / m \\ \frac{I_{xz} M_{\text{ext},z} + I_{zz} M_{\text{ext},x} + (I_{xx} I_{xz} - I_{xz} I_{yy} + I_{zx} I_{zz}) p q - (I_{xz}^2 + I_{zz}^2 - I_{yy} I_{zz}) q r}{\frac{I_{xx} I_{zz} - I_{xz} I_{zx}}{M_{\text{ext},y} - (I_{xx} - I_{zz}) p r - I_{zx} p^2 + I_{xz} r^2}} \\ \frac{I_{yy}}{I_{xx} M_{\text{ext},z} + I_{zx} M_{\text{ext},x} + (I_{xx}^2 + I_{zz}^2 - I_{xx} I_{yy}) p q + (I_{zx} I_{yy} - I_{xz} I_{xx} - I_{zx} I_{zz}) q r} \\ \frac{I_{xx} I_{zz} - I_{xz} I_{zx}}{I_{xx} I_{zz} - I_{xz} I_{zx}} \end{pmatrix} \quad (113)$$

APPENDIX C
INTERNATIONAL STANDARD ATMOSPHERE

$$k = -0.0065 \frac{K}{m} \quad (114)$$

$$P_0 = 101325 Pa \quad (115)$$

$$T_0 = 288.15 K \quad (116)$$

$$R = 287 \frac{J}{kg K} \quad (117)$$

$$\gamma = 1.4 \quad (118)$$

$$k = -0.0065 \frac{K}{m} \quad (119)$$

$$T(h \leq 11 km) = T_0 + h k \quad (120)$$

$$T(11 km < h < 20 km) = T_0 + k * 11000; \quad (121)$$

$$P(h \leq 11 km) = P_0 \left(\frac{T}{T_0} \right)^{\frac{-g}{Rk}} \quad (122)$$

$$P(11 km < h < 20 km) = P_{11} e^{-\frac{g(h-h_{11})}{RT_{11}}} \quad (123)$$

$$\rho_{air} = \frac{P}{RT} \quad (124)$$

$$v_{sound} = \sqrt{\gamma RT} \quad (125)$$

$$M = \frac{v_{TAS}}{v_{sound}} \quad (126)$$

$$a_0 = \sqrt{\gamma RT_0} \quad (127)$$

$$\Delta P = P \left(\left(\frac{(\gamma - 1)}{2} M^2 + 1 \right)^{\frac{\gamma}{\gamma - 1}} - 1 \right) \quad (128)$$

$$v_{CAS} = \sqrt{\frac{2a_0^2}{\gamma - 1} \left(\left(\frac{\Delta P}{P_0} + 1 \right)^{\frac{\gamma - 1}{\gamma}} - 1 \right)} \quad (129)$$

$$v_{TAS} = \sqrt{u^2 + v^2 + w^2} \quad (130)$$

$$\alpha = \text{atan2}(w, u) \quad (131)$$

$$\beta = \text{atan2}(v, v_{TAS}) \quad (132)$$

APPENDIX D AERODYNAMIC MODELING

$$C_L = C_{L_0} + C_{L_\alpha} \alpha + C_{L_{\delta_e}} \delta_e + \frac{C_{L_{\dot{\alpha}}} \dot{\alpha} c}{2 v_{TAS}} + \frac{C_{L_q} q c}{2 v_{TAS}} \quad (133)$$

$$C_D = C_{D_0} + C_{D_\alpha} \alpha + C_{D_{\delta_e}} \delta_e \quad (134)$$

$$C_Y = C_{Y_\beta} \beta + C_{Y_{\delta_a}} \delta_a + C_{Y_{\delta_r}} \delta_r + \frac{C_{Y_p} p b}{2 v_{TAS}} + \frac{C_{Y_r} r b}{2 v_{TAS}} \quad (135)$$

$$C_l = C_{l_\beta} \beta + C_{l_{\delta_a}} \delta_a + C_{l_{\delta_r}} \delta_r + \frac{C_{l_p} p b}{2 v_{TAS}} + \frac{C_{l_r} r b}{2 v_{TAS}} \quad (136)$$

$$C_m = C_{m_0} + C_{m_\alpha} \alpha + C_{m_{\delta_e}} \delta_e + \frac{C_{m_{\dot{\alpha}}} \dot{\alpha} c}{2 v_{TAS}} + \frac{C_{m_q} q c}{2 v_{TAS}} \quad (137)$$

$$C_n = C_{n_\beta} \beta + C_{n_{\delta_a}} \delta_a + C_{n_{\delta_r}} \delta_r + \frac{C_{n_p} p b}{2 v_{TAS}} + \frac{C_{n_r} r b}{2 v_{TAS}} \quad (138)$$

$$C_{X_b} = C_L \sin(\alpha) - C_D \cos(\alpha) \quad (139)$$

$$C_{Y_b} = C_Y \quad (140)$$

$$C_{Z_b} = -C_L \cos(\alpha) - C_D \sin(\alpha) \quad (141)$$

$$C_{l_b} = C_l \cos(\alpha) - C_n \sin(\alpha) \quad (142)$$

$$C_{m_b} = C_m \quad (143)$$

$$C_{n_b} = C_n \cos(\alpha) + C_l \sin(\alpha) \quad (144)$$

$$Q = \frac{1}{2} \rho_{\text{air}} v_{TAS}^2 \quad (145)$$

$$F_{\text{aero},x} = C_{X_b} Q S \quad (146)$$

$$F_{\text{aero},y} = C_{Y_b} Q S \quad (147)$$

$$F_{\text{aero},z} = C_{Z_b} Q S \quad (148)$$

$$M_{\text{aero},x} = C_{l_b} Q S b - F_{\text{aero},y} z_{CG} - F_{\text{aero},z} y_{CG} \quad (149)$$

$$M_{\text{aero},y} = C_{m_b} Q S c + F_{\text{aero},x} z_{CG} - F_{\text{aero},z} c (CG_{\text{mac}} - 0.25) \quad (150)$$

$$M_{\text{aero},z} = C_{n_b} Q S b + F_{\text{aero},x} y_{CG} + F_{\text{aero},y} c (CG_{\text{mac}} - 0.25) \quad (151)$$

APPENDIX E PROPULSION MODELING

$$T_{\text{linear}} = T = \delta_t T_{\text{max}} \quad (152)$$

$$T_{\text{nonlinear}} = T = \delta_t T_{\text{max}} \left(\frac{v_{TAS}}{v_{\text{ref}}} \right)^{n_v} \left(\frac{\rho_{\text{air}}}{\rho_{\text{ref}}} \right)^{n_\rho} \quad (153)$$

$$\vec{F}_{\text{thrust}} = \begin{pmatrix} F_{\text{thrust},x} \\ F_{\text{thrust},y} \\ F_{\text{thrust},z} \end{pmatrix} = \begin{pmatrix} T \cos(\alpha_F) \\ 0 \\ T \sin(\alpha_F) \end{pmatrix} \quad (154)$$

$$\vec{M}_{\text{thrust}} = \begin{pmatrix} M_{\text{thrust},x} \\ M_{\text{thrust},y} \\ M_{\text{thrust},z} \end{pmatrix} = \begin{pmatrix} 0 \\ T \cos(\alpha_F) Z_F - T \sin(\alpha_F) X_F \\ 0 \end{pmatrix} \quad (155)$$

APPENDIX F

TRIMMED CONDITION: LEVEL UNACCELERATED FLIGHT AT 5000 FT

$$\mathbf{x}_0 = (x_0, y_0, z_0, \phi_0, \theta_0, \psi_0, p_0, q_0, r_0, u_0, v_0, w_0)^T \quad (156)$$

$$\dot{\mathbf{x}}_0 = f(\mathbf{x}_0) = f(x_0, y_0, z_0, \phi_0, \theta_0, \psi_0, p_0, q_0, r_0, u_0, v_0, w_0) = 0 \quad (157)$$

$$\mathbf{y}_0 = (x_0, y_0, z_0, \phi_0, \theta_0, \psi_0, u_0, v_0, w_0, p_0, q_0, r_0, \alpha_0, \beta_0, \gamma_0, \zeta_0, v_{\text{TAS}_0}, \rho_0)^T \quad (158)$$

$$\mathbf{x}_0 = (0, 0, -1524, 0, 0, 0, 62.3866, 0, 0, 0, 0, 0)^T \quad (159)$$

$$\mathbf{u}_0 = (\delta_{e_0}, \delta_{a_0}, \delta_{r_0}, \delta_{t_0})^T = (-0.0032115, 0, 0, 0.6792)^T \quad (160)$$

$$\mathbf{y}_0 = (0, 0, -1524, 0, 0, 0, 62.3866, 0, 0, 0, 0, 0, 0, 0, 0, 62.3866, 1.0557)^T \quad (161)$$

APPENDIX G

LINEARIZED STATES DIFFERENTIAL EQUATIONS

$$\mathbf{x} = \mathbf{x}_0 + \Delta \mathbf{x} = \begin{pmatrix} x_0 + \Delta x \\ y_0 + \Delta y \\ z_0 + \Delta z \\ \phi_0 + \Delta \phi \\ \theta_0 + \Delta \theta \\ \psi_0 + \Delta \psi \\ u_0 + \Delta u \\ v_0 + \Delta v \\ w_0 + \Delta w \\ p_0 + \Delta p \\ q_0 + \Delta q \\ r_0 + \Delta r \end{pmatrix} \quad (162)$$

$$\mathbf{u} = \mathbf{u}_0 + \Delta \mathbf{u} = \begin{pmatrix} \delta_{e_0} + \Delta \delta_e \\ \delta_{a_0} + \Delta \delta_a \\ \delta_{r_0} + \Delta \delta_r \\ \delta_{t_0} + \Delta \delta_t \end{pmatrix} \quad (163)$$

$$\mathbf{y} = \mathbf{y}_0 + \Delta \mathbf{y} = \begin{pmatrix} x_0 + \Delta x \\ y_0 + \Delta y \\ z_0 + \Delta z \\ \phi_0 + \Delta \phi \\ \theta_0 + \Delta \theta \\ \psi_0 + \Delta \psi \\ u_0 + \Delta u \\ v_0 + \Delta v \\ w_0 + \Delta w \\ p_0 + \Delta p \\ q_0 + \Delta q \\ r_0 + \Delta r \\ \alpha_0 + \Delta \alpha \\ \beta_0 + \Delta \beta \\ \gamma_0 + \Delta \gamma \\ \zeta_0 + \Delta \zeta \\ v_{\text{TAS}_0} + \Delta v_{\text{TAS}} \\ \rho_0 + \Delta \rho \end{pmatrix} \quad (164)$$

APPENDIX H

AIRCRAFT LONGITUDINAL DYNAMICS

$$\Delta \dot{x} = \Delta u \quad (171)$$

$$\Delta \dot{z} = \Delta w - 62.39 \Delta \theta \quad (172)$$

$$\Delta \dot{\theta} = \Delta q \quad (173)$$

$$\Delta \dot{u} = -0.0001 \Delta z - 9.807 \Delta \theta - 0.0477 \Delta u + 0.2388 \Delta w + 1.91 \Delta \delta_e + 1.462 \Delta \delta_t \quad (174)$$

$$\Delta \dot{w} = -0.0022 \Delta z - 0.3152 \Delta u - 2.64 \Delta w + 60.9 \Delta q - 13.69 \Delta \delta_e + 0.0255 \Delta \delta_t \quad (175)$$

$$\Delta \dot{q} = 0.0005 \Delta u - 0.2494 \Delta w - 3.971 \Delta q - 33.99 \Delta \delta_e - 0.0146 \Delta \delta_t \quad (176)$$

$$\Delta \mathbf{x} = (\Delta x, \Delta z, \Delta \theta, \Delta u, \Delta w, \Delta q)^T \quad (177)$$

$$\Delta \mathbf{u} = (\Delta \delta_e, \Delta \delta_t)^T \quad (178)$$

$$\Delta y = \Delta \theta \quad (179)$$

$$\Delta \dot{\mathbf{x}} = A_{\text{long}} \Delta \mathbf{x} + B_{\text{long}} \Delta \mathbf{u} \quad (180)$$

$$\Delta y = C_{\text{long}} \Delta \mathbf{x} + D_{\text{long}} \Delta \mathbf{u} \quad (181)$$

$$A_{\text{long}} = \begin{pmatrix} 0 & 0 & 0 & 1.0 & 0 & 0 \\ 0 & 0 & -62.39 & 0 & 1.0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.0 \\ 0 & -0.0001 & -9.807 & -0.0477 & 0.2388 & 0 \\ 0 & -0.0022 & 0 & -0.3152 & -2.64 & 60.9 \\ 0 & 0 & 0 & 0.0005 & -0.2494 & -3.971 \end{pmatrix} \quad (182)$$

$$B_{\text{long}} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1.91 & 1.462 \\ -13.69 & 0.0255 \\ -33.99 & -0.0146 \end{pmatrix} \quad (183)$$

$$C_{\text{long}} = \begin{pmatrix} 0 & 0 & 1.0 & 0 & 0 & 0 \end{pmatrix} \quad (184)$$

$$D_{\text{long}} = \begin{pmatrix} 0 & 0 \end{pmatrix} \quad (185)$$

APPENDIX I AIRCRAFT LATERAL DYNAMICS

$$\Delta \dot{y} = 62.39 \Delta \psi + \Delta v \quad (186)$$

$$\Delta \dot{\phi} = \Delta p \quad (187)$$

$$\Delta \dot{\psi} = \Delta r \quad (188)$$

$$\Delta \dot{v} = 9.807 \Delta \phi - 0.1582 \Delta v - 0.103 \Delta p - 61.8 \Delta r - 5.953 \Delta \delta_r \quad (189)$$

$$\Delta \dot{p} = -0.3765 \Delta v - 11.57 \Delta p + 2.272 \Delta r - 50.19 \Delta \delta_a + 3.178 \Delta \delta_r \quad (190)$$

$$\Delta \dot{r} = 0.137 \Delta v - 0.3595 \Delta p - 1.159 \Delta r - 7.202 \Delta \delta_a - 8.754 \Delta \delta_r \quad (191)$$

$$\Delta \mathbf{x} = (\Delta y, \Delta \phi, \Delta \psi, \Delta v, \Delta p, \Delta r)^T \quad (192)$$

$$\Delta \mathbf{u} = (\Delta \delta_a, \Delta \delta_r)^T \quad (193)$$

$$\Delta y = \Delta \psi \quad (194)$$

$$\Delta \dot{\mathbf{x}} = A_{\text{lat}} \Delta \mathbf{x} + B_{\text{lat}} \Delta \mathbf{u} \quad (195)$$

$$\Delta y = C_{\text{lat}} \Delta \mathbf{x} + D_{\text{lat}} \Delta \mathbf{u} \quad (196)$$

$$A_{\text{lat}} = \begin{pmatrix} 0 & 0 & 62.39 & 1.0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.0 \\ 0 & 9.807 & 0 & -0.1582 & -0.103 & -61.8 \\ 0 & 0 & 0 & -0.3765 & -11.57 & 2.272 \\ 0 & 0 & 0 & 0.137 & -0.3595 & -1.159 \end{pmatrix} \quad (197)$$

$$B_{\text{lat}} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & -5.953 \\ -50.19 & 3.178 \\ -7.202 & -8.754 \end{pmatrix} \quad (198)$$

$$C_{\text{lat}} = \begin{pmatrix} 0 & 0 & 1.0 & 0 & 0 & 0 \end{pmatrix} \quad (199)$$

$$D_{\text{lat}} = \begin{pmatrix} 0 & 0 \end{pmatrix} \quad (200)$$