Loads Alleviation on an Airfoil via Reinforcement Learning

Esteban Hufstedler* and Philippe Chatelain†
Institute of Mechanics, Materials and Civil Engineering, Université Catholique de Louvain (UCL), Louvain-la-Neuve, Belgium

Reinforcement learning techniques propose an alternative to the potentially expensive use of model predictive control. This research uses reinforcement learning techniques to develop a control policy to reduce the unsteady turbulent loads on a 2D airfoil. The learned control strategy compares well with the classical linear-quadratic-Gaussian control, when applied to the turbulence. When tested with a discrete gust, the learned controller performs relatively poorly.

Nomenclature

$\mathbf{A}, \mathbf{A_d}, \mathbf{A_e}$	=	Continuous time, discrete time, and extended state evolution matrices
$\mathbf{B}, \mathbf{B_d}, \mathbf{B_e}$	=	Continuous time, discrete time, and extended control matrices
$\mathbf{B'}, \mathbf{B'_d}, \mathbf{B'_e}$	=	Continuous time, discrete time, and extended turbulence generation matrices
$\mathbf{C}$	=	Sensing matrix
C_L	=	Lift coefficient, lift / (dynamic pressure * chord length)
$\mathbf{I}$	=	Identity matrix
L_w	=	Turbulence length scale, m
Q	=	Action-value function
$\mathbf{Q}$	=	State cost matrix
U	=	Freestream velocity, m/s
c	=	Chord length of the airfoil, m
$f_()$	=	Constants related to the flap hinge position
i_e	=	Episode index
n_e	=	Total number of episodes
u	=	Control input: acceleration of the flap, m/s
w	=	Velocity perturbations normal to the freestream, m/s
$\mathbf{x}$	=	State of the airfoil system
$\mathbf{y}$	=	Observation of the airfoil system
$z_()$	=	Parameters to model the wake's influence on the airfoil
α	=	Angle of attack, radians
β	=	Flap deflection angle, radians
β_{max}	=	Maximum flap deflection, radians
$\dot{\beta}_{max}$	=	Maximum flap deflection speed, radians$/s$
$\ddot{\beta}_{max}$	=	Maximum flap deflection acceleration, radians$/s^2$
ϵ	=	Reinforcement learning action-choice parameter
σ_w	=	Turbulence intensity, m/s
Δt	=	Simulation time step duration, s
$\dot{()}$	=	Derivative with respect to time
$\ddot{()}$	=	Second derivative with respect to time

I. Introduction

This is an attempt to alleviate the time-varying loads on an airfoil in turbulent flow using a reinforcement-learned policy to dictate its flap motion, rather than classical techniques such as model predictive control. The system in

*Post-doctoral researcher, Department of Thermodynamics and Fluid Mechanics, esteban.hufstedler@uclouvain.be.
†Professor, Department of Thermodynamics and Fluid Mechanics, philippe.chatelain@uclouvain.be.

question is an airfoil with a flap, exposed to oncoming normal-velocity perturbations. The airfoil's flap is limited: its deflection, speed, and acceleration are limited in magnitude. The controllers sense the instantaneous flap position, speed, and some oncoming perturbations, and apply an acceleration to the flap. As a baseline, the learned policies are compared to linear-quadratic-Gaussian controllers.

Rejection of loads due to atmospheric unsteadiness is a lively topic of research. Regan and Jutte[1] describe a range of benefits of loads alleviation: reduction in structural weight due to reduction of loads, reduced fatigue of wings, improved ride quality, and more. These approaches typically involve measurements of local quantities, but some research includes observations of the upstream flow conditions. The von Kármán turbulence spectrum[2] is often used to model such unsteadiness, providing an input to simulations of gust response. Another popular type of unsteadiness is the discrete one-minus-cosine gust[3].

find more examples of load rejection?

Thin airfoil theory[4] provides relatively simple methods of modeling high-Reynolds number flow around an airfoil. This assumes inviscid, attached flow over the airfoil, and a wake that extends from the trailing edge. In unsteady thin airfoil theory, the wake evolves over time in response to the motion of the airfoil or oncoming velocity perturbations. A key result of the unsteadiness is the asymptotic approach to, rather than instantaneous achievement of, the steady-state lift coefficient. The Wagner[5] and Küssner[6] functions were developed to describe these unsteady effects. Leishman[7] developed a state-space model of these functions, allowing for more efficient modeling of unsteady thin airfoil theory. This model included a set of wake-capturing parameters, yielding a model of the time-varying lift coefficient, C_L.

The baseline controller in this study used linear-quadratic-Gaussian (LQG) control. This classical approach optimally controls a system with limited observations, including the possibility that the system may be perturbed by normally-distributed noise.

Reinforcement learning[8] (RL) is an approach for generating control strategies based on interaction with an environment. The learning agent performs actions in the environment, and is given a numerical reward. The goal is for the agent to maximize its cumulative reward. The agent develops a 'policy' which determines what actions it will perform. It initially knows nothing of the environment, and must explore the environment (by performing new actions with unknown rewards) and exploit it (performing known, rewarding actions). This is a technique of learning by trial and error, rather than mathematical analysis. As such, it can discover control methods for systems that are difficult to mathematically analyze. This may be useful for controlling systems that are nonlinear or limited in their actuation.

This paper uses Q-learning; a type of RL where an agent attempts to learn how valuable each possible action is from each state it has observed. This results in the action-value function, $Q(s, a)$, that maps a given state, s, and action, a, to a numerical value. At each timestep, a RL agent must observe its environment, choose an action, perform the action, and then obtain a reward and make a new observation of the environment. From this sequence of observation, action, reward, and observation, it updates its estimate of Q. Here, the Q function was learned with the Double Deep-Q Network[9] (DDQN) approach. The basic deep Q network approach uses a multilayer artificial neural network to approximate the Q function and choose its actions. The DDQN improves on this by using two separate networks: the evaluation network is trained over time and is used to choose actions, and the target network is used to estimate the value of the actions. Periodically, the properties of the evaluation network are copied to the target network. This separation and periodic replacement makes the system less overoptimistic, resulting in better performance. This algorithm includes 'experience replay', where the RL agent maintains a buffer of previous experiences, and periodically trains itself on a randomly selected set of these memories. In this study, the agent chose actions via the epsilon-greedy method: it picked the most rewarding action with probability $1 - \epsilon$, or a uniformly random action with probability ϵ.

This paper is organized in the following manner. First, the aerodynamic model of the airfoil is presented in section II. The different control schemes are introduced in section III. Section IV describes the simulation parameters, and section V presents the results.

II. Numerical Model of the Airfoil System

The airfoil, seen in Figure 1, is modeled with unsteady thin airfoil theory. It has chord length c, steady angle of attack α, and time-varying flap deflection β. It can be controlled by applying an angular acceleration $\ddot{\beta}$. The flap is limited to deflections of $|\beta| \leq \beta_{max}$, speeds of $|\dot{\beta}| \leq \dot{\beta}_{max}$, and accelerations of $|\ddot{\beta}| \leq \ddot{\beta}_{max}$. If the flap reaches its angular maximum, it stops suddenly.

The streamwise velocity, U, is constant. The oncoming flow carries vertical velocity perturbations, $w(t)$, which cause unwanted fluctuations in the lift. This turbulence is generated using white noise applied to transfer functions which approximate the von Kármán turbulence spectrum, with length scale L_w and intensity σ_w. This turbulence is 'frozen:' it does not evolve as it convects downstream and interacts with the airfoil. The unsteady effects of the wake are

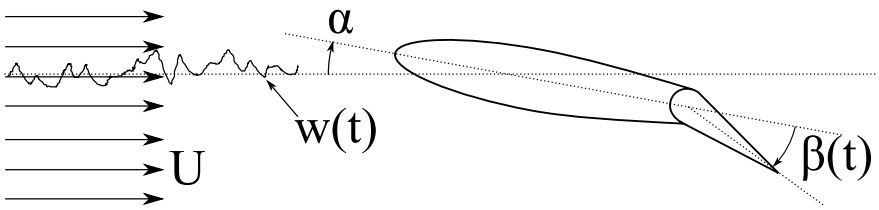


Fig. 1 Flapped airfoil in freestream, with incoming velocity perturbations.

captured, however, using Leishman's indicial approach of integrating wake-capturing parameters, which are denoted $z_{()}$.

A. Continuous-time evolution

The state of the system is

$$\mathbf{x} = \begin{bmatrix} \beta & \dot{\beta} & z_{wag,1} & z_{wag,2} & z_{kus,1} & z_{kus,2} & C_L & w \end{bmatrix}^T, \tag{1}$$

which evolves as

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\ddot{\beta}(t) + \mathbf{B}'\dot{w}(t), \tag{2}$$

where $\ddot{\beta}(t)$ is the applied control, which accelerates the flap angle. The matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{B}'$ are defined in the Appendix. With steady conditions, the lift coefficient simplifies to

$$C_L = 2\pi\left(\alpha + \frac{w}{U}\right) + 2f_{10}\beta, \tag{3}$$

where f_{10} is a constant (based on the flap length) which is defined in the Appendix.

1. Discrete-time evolution

The system was simulated in discrete time steps of length Δt, which enables us to easily include observations of the oncoming flow perturbations. The discrete-time evolution was approximated using Euler integration from timestep i to $i+1$ as

$$\mathbf{x}_{i+1} = (\mathbf{I} + \mathbf{A}\Delta t)\mathbf{x}_i + (\mathbf{B}\Delta t)\ddot{\beta}_i + (\mathbf{B}'\Delta t)\dot{w}_i, \tag{4}$$

$$= \mathbf{A_d}\mathbf{x_i} + \mathbf{B_d}\ddot{\beta}_i + \mathbf{B'_d}\dot{w}_i. \tag{5}$$

To include observations of the upstream turbulence, these matrices are extended to include N upcoming velocity perturbations:

$$\begin{bmatrix} \mathbf{x} \\ w_{+1} \\ \vdots \\ w_{+N} \end{bmatrix}_{i+1} = \begin{bmatrix} \mathbf{A_{d'}} & 0 \\ 0 & \mathbf{I} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ w_{+1} \\ \vdots \\ w_{+N} \end{bmatrix}_i + \begin{bmatrix} \mathbf{B_d} \\ 0 \end{bmatrix} \ddot{\beta}_i + \begin{bmatrix} 0 \\ \vdots \\ 1 \end{bmatrix} w_{new,i}, \tag{6}$$

$$\mathbf{x_e}_{,i+1} = \mathbf{A_e}\mathbf{x_e}_{,i} + \mathbf{B_e}\ddot{\beta}_i + \mathbf{B'_e}w_{new,i}, \tag{7}$$

where $\mathbf{A_{d'}}$ is $\mathbf{A_d}$, but with the lower-right entry set equal to zero. The $w_{new,i}$ is the newest value of the upstream turbulence.

The controllers do not directly access the state, and instead observe $\mathbf{y}$:

$$\mathbf{y} = \mathbf{C}\mathbf{x_e}, \tag{8}$$

where $\mathbf{C}$ extracts β, $\dot{\beta}$, C_L, and some number of the oncoming velocity perturbations.

III. Control methods

Three control methods were examined: LQG control of the full system, LQG control of a simplified quasi-steady version of the system (LQG-QS), and the RL approach. The RL control was limited to three values of actuation, compared to the continuous control available to the LQG approaches.

A. Linear-Quadratic-Gaussian Control

This unsteady airfoil model is linear time-invariant, which simplifies the analysis. Applying LQG control to the extended system gives the feedback gain, which defines the control. In this study, this was approximated through iteration of the forward and backward Riccati equations. The state cost matrix $\mathbf{Q}$ penalizes the lift coefficient error $(C_L - C_L(0))^2$ with a value of 1, and the use of control, $\ddot{\beta}^2$ with a value of 5. The state noise and process noise covariance matrices are 10^{-4} times the identity matrix. If the LQG controller commands a $|\ddot{\beta}| > \ddot{\beta}_{max}$, the system will only accelerate at $\ddot{\beta}_{max}$.

To examine the importance of wake delay effects, LQG was also performed using system matrices corresponding to steady thin airfoil theory. This is referred to as quasi-steady control, or LQG-QS.

B. Reinforcement-Learned Control

This implementation of Q-learning allowed for three discrete actions: no flap acceleration, or accelerating the flap by $\pm\ddot{\beta}_{max}$. To mimic the information retained by the Kalman filter, the RL state contains the observed β, $\dot{\beta}$, C_L, and velocity perturbations of the 50 previous timesteps. To encourage exploration at the start of the learning process and exploitation of rewarding behaviors at the end, the value of ϵ was chosen to decrease from ϵ_0=0.5 to ϵ_{final}=0.01. The value at episode number i_e is

$$\epsilon = \epsilon_0 \exp\left(-\frac{i_e}{n_e}\log\left(\frac{\epsilon_0}{\epsilon_{final}}\right)\right), \tag{9}$$

where n_e is the maximum number of episodes.

Two types of agents were trained, with different reward functions,

$$-\left|C_L - C_L(0)\right|^p, \tag{10}$$

where p is 1 or 2. The power of two emphasizes reduction of large deviations. These are referred to as RL Agents 1 and 2.

The DDQN network involves a number of parameters. At each timestep, an agent stored its observations in the memory buffer, which could hold 1e5 observations. The networks were implemented in Keras[10] as two dense layers of 32 neurons each. The evaluation network was trained every 50 timesteps using an Adam optimizer with learning rate 1e-4, discounting parameter of 0.9, batch size of 32. The evaluation network was copied to the training network every 100 timesteps. Training was performed using an Adam optimizer.

When the agents were evaluated, ϵ was set to zero to ensure deterministic action choices.

IV. Simulation Parameters

The freestream speed, chord length, and flap length were modeled on those of a Piper Cub light aircraft flying near its stall speed of 16 km/h, with a chord length of 1.54 m and a flap length of 24%. The flap deflection is limited to ± 10° to maintain approximately attached flow. The maximum flap deflection speed is 20°/s, and maximum flap acceleration is $160°/s^2$. The turbulence intensity and length scales were modeled on von Kármán turbulence at an altitude of 1000 ft[11].

Since this is a model of inviscid and incompressible flow, the freestream velocity and chord length were set equal to unity in the simulation, and other parameters were nondimensionalized by the physical chord length and freestream speed. These parameters are listed in Table 1.

The RL agents were trained over the course of 1000 episodes of 4000 time steps each, totaling $4\text{x}10^6$ timesteps. Each training episode had a random σ_w between 0.46 and 1.34 U. The effectiveness of each control scheme was tested with episodes of 10000 time steps, or 1000 convective time units, for each evaluated σ_w.

Parameter	Value(s) or range
Velocity sensing distance upstream	$5c$
Δt	$0.1\ c/U$
Training Episodes	1000
Training episode length	4000 time steps
Evaluation episode length	10000 time steps
Training σ_w	0.46 to 1.34 U
L_w	200 c
β_{max}	10°
$\dot{\beta}_{max}$	$1.8\ ^\circ\ U/c$
$\ddot{\beta}_{max}$	$1.3\ ^\circ\ (U/c)^2$

Table 1 Simulation parameters

V. Results

A series of simulations were performed. First, the effect of different sensing positions was examined. Second, the amount of saturation of the LQG controller was found. Next, the RL agents were trained to control the turbulent lift fluctuations, and their effectiveness compared to the LQG controllers. Finally, these controllers were applied to the one-minus-cosine gust scenario.

A. Effect of sensing positions

The LQG and LQG-QS controllers were applied in a weakly turbulent flow to examine the importance of sensor positions. Two types of tests were conducted: 'dense' sensing where the perturbations were measured at every spacing of $U\Delta t$ until a maximum position, and sensing which only measured at that furthest point. Figure 2 displays the effectiveness of these sensing methods for a range of furthest points, presented as the root mean square (RMS) of the controlled C_L, normalized by the RMS of the uncontrolled C_L.

Both control techniques reduced the variation in lift, but the inclusion of the unsteady state variables in the full LQG yielded superior results. Looking at the LQG results, it was useful to sense upstream of the airfoil, but with diminishing returns. In this case, it was not useful to have sensors beyond $5c$ upstream. This length is related to how long it takes to deflect the flap, rather than needing to know about distant turbulence. Additionally, since the turbulence was frozen, having one sensor was nearly as useful as having many. As a result, the rest of the simulations in this paper use a single sensor 5c upstream of the airfoil.

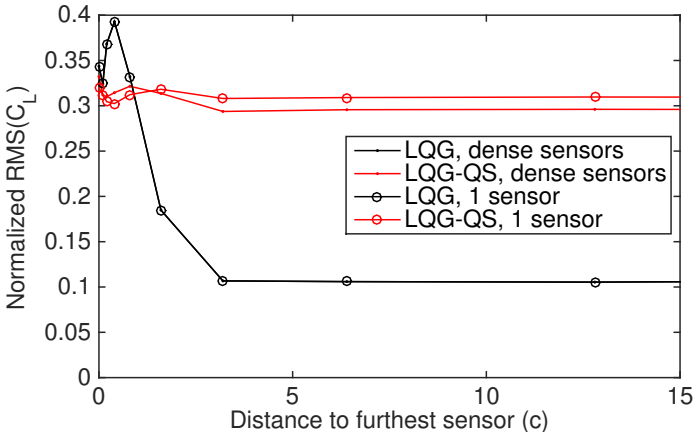


Fig. 2 Effect of number of sensing distance and number of sensors. Turbulence intensity $\sigma_w = 0.4$

B. Saturation of LQG controller

Since the available control effort was limited, the LQG controller would attempt to command impossible accelerations in response to strong turbulence. Figure 3 shows the relative frequency of these impossible commands. The controller was saturated nearly 20% of the time with the strongest tested turbulence.

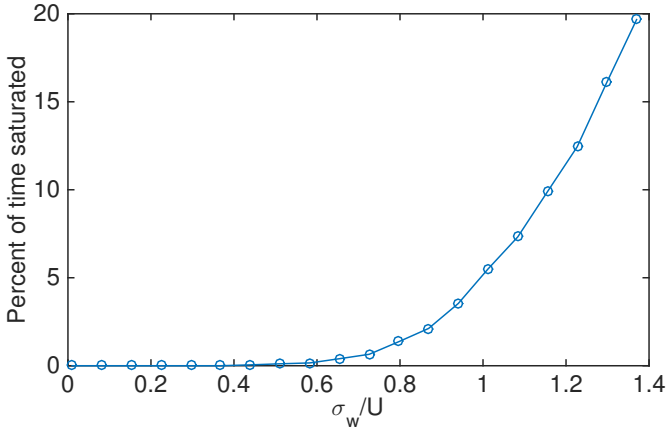
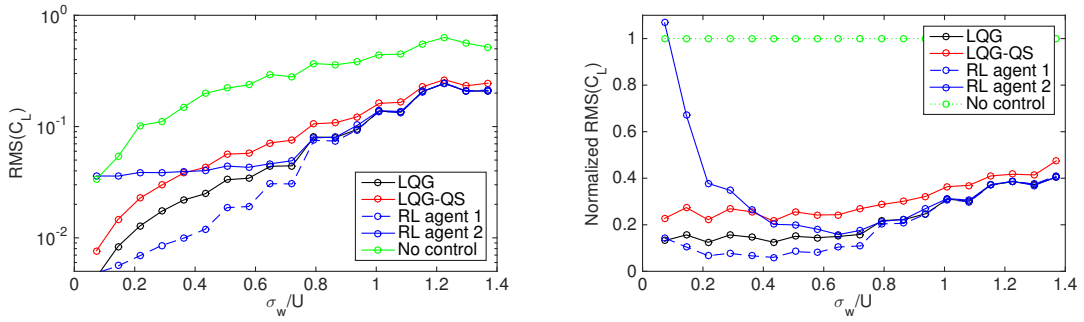


Fig. 3 Percent of time the LQG controller was saturated

C. Reinforcment Learning Results

The results of the different control techniques are shown in Figure 4. The absolute RMS of C_L is plotted in Figure 4a, and Figure 4b is normalized by the no-control RMS. Above σ_w=0.8, the RL agents performed as well as the LQG controller. Below that intensity, the three approaches diffeedr. RL agent 2, with a reward function that emphasizes rejecting large disturbances, did not learn to control weak turbulence well. RL agent 1, in contrast, reduced the variation in C_L even more than the LQG controller, until the disturbances were extremely weak. This is likely because the use of control was penalized with the LQG controller, but not with the RL agent.



(a) RMS of C_L. **(b) RMS of C_L, normalized by the no-control case.**

Fig. 4 RMS of C_L, with and without normalization, for variable σ_w. Note log and linear y-axes.

D. Discrete gusts

The effect of a strong discrete one-minus-cosine gust is shown in Figure 5. The LQG controller was the most effective, reducing the magnitude of lift variation by 90%. The worst was RL agent 1, which reduced the peak magnitude by 19%. The quadratic RL agent reduced the lift variation by 48%. The relative ineffectiveness of the RL agents demonstrates the risk of applying RL control to situations they have not been trained for. Also, both RL agents show some sawtooth-like behavior, as the discrete actions make it difficult to smoothly follow a curve.

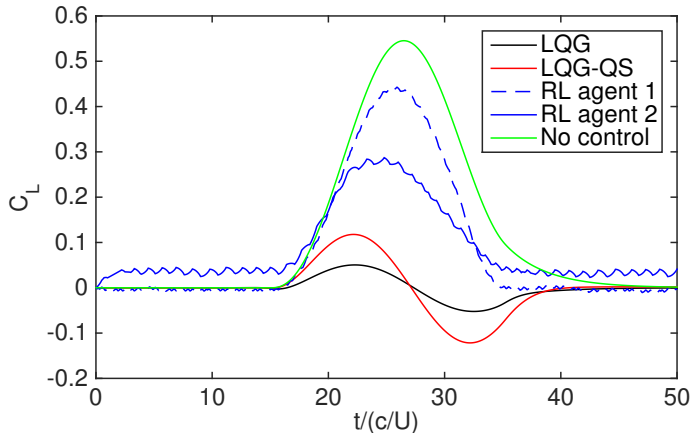


Fig. 5 C_L response to a one-minus-cosine gust, velocity amplitude $0.1U$

VI. Conclusion

Reinforcement learning provides a means to generate effective control policies without understanding the dynamics of the system. Application of such techniques successfully reduced lift fluctuations on a simple model of an airfoil in von Kármán turbulence, with performance comparable to LQG control. Unfortunately, the RL control methods were less effective when applied to circumstances beyond their training.

Future extensions of this work may involve two main thrusts: improving the system model, or improving the learning techniques. To make the system more physically relevant, the sensing could model LiDAR (light detection and ranging) instead of point measurements, and include sensor noise. The turbulence could also be modified, such that it is two-dimensional or evolves over time. A further elaboration could include pitching and heaving of the airfoil, as it attempts to avoid patches of upcoming turbulence. More advanced RL strategies would be useful, such as those that allow for continuous values of control input.

Appendix

This appendix provides the values of the state evolution matrices, based on Leishman's approach. The $f_()$ parameters depend on the position of the flap hinge. With x_h as the distance from the leading edge of the airfoil to the flap:

$$e = 2\frac{x_h}{c} - 1, \tag{11}$$

$$f_4 = e\sqrt{1-e^2} - \arccos(e), \tag{12}$$

$$f_{10} = \sqrt{1-e^2} + \arccos(e), \tag{13}$$

$$f_{11} = (1-2e)\arccos(e) + (2-e)\sqrt{1-e^2}. \tag{14}$$

The state evolution matrices are

$$\frac{d}{dt}\begin{bmatrix}\beta\\ \dot{\beta}\\ z_{wag,1}\\ z_{wag,2}\\ z_{kus,1}\\ z_{kus,2}\\ C_L\\ w\end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0\\ \frac{f_{10}}{\pi} & \frac{cf_{11}}{4\pi U} & -0.07419\frac{U^2}{c^2} & -0.7774\frac{U}{c} & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 0 & -1.004\frac{U^2}{c^2} & -3.883\frac{U}{c} & 0 & \frac{1}{U}\\ 0.4388\frac{f_{10}U}{c} & f_{10}+0.1097f_{11} & -0.1023\frac{U^3}{c^3} & -0.8387\frac{U^2}{c^2} & -10.59\frac{U^3}{c^3} & -34.62\frac{U^2}{c^2} & 0 & 10.54\frac{1}{c}\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\end{bmatrix}\begin{bmatrix}\beta\\ \dot{\beta}\\ z_{wag,1}\\ z_{wag,2}\\ z_{kus,1}\\ z_{kus,2}\\ C_L\\ w\end{bmatrix} \ldots \tag{15}$$

$$+\begin{bmatrix}0\\1\\0\\0\\0\\0\\-f_4\frac{c}{2}+\frac{cf_{11}}{4U}\\0\end{bmatrix}\ddot{\beta}(t)+\begin{bmatrix}0\\0\\0\\0\\0\\0\\0\\1\end{bmatrix}\dot{w}(t), \tag{16}$$

$$\mathbf{x} = \mathbf{A}\mathbf{x} + \mathbf{B}\ddot{\beta}(t) + \mathbf{B}'\dot{w}(t), \tag{17}$$

The $z_{()}$ parameters are initialized as if the airfoil has been sitting at a steady state for infinite time:

$$z_{wag,1}(0) = 4.290\frac{c^2(\pi\alpha + f_{10}\beta(0))}{U^2} \tag{18}$$

$$z_{wag,2}(0) = 0 \tag{19}$$

$$z_{kus,1}(0) = 0.9959\frac{c^2 w(0)}{U^3} \tag{20}$$

$$z_{kus,2}(0) = 0 \tag{21}$$

$$\tag{22}$$

Acknowledgments

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement No 725627).

References

[1] "Survey of Applications of Active Control Technology for Gust Alleviation and New Challenges for Lighter-weight Aircraft," Tech. Rep. 216008, NASA, 2012.

[2] "MIL-STD-1797A: Flying Qualities of Piloted Aircraft," Tech. rep., US Department of Defense, 1997.

[3] "Advisory circular 25.341-1: Dynamic gust loads," Tech. rep., US Federal Aviation Administration, 2014.

[4] Fung, Y., *An introduction to the theory of aeroelasticity*, Courier Corporation, 2002.

[5] Wagner, H., "Über die Entstehung des dynamischen Auftriebes von Tragflügeln," *Zeitschrift für Angewandte Mathematik und Mechanik*, Vol. 5, No. 1, 1925, pp. 17–35.

[6] Küssner, H. G., "Zusammenfassender Bericht über den instationären Auftrieb von Flügeln," *Luftfahrtforschung*, Vol. 13, No. 12, 1936, pp. 410–424.

[7] Leishman, J. G., "Unsteady lift of a flapped airfoil by indicial concepts," *Journal of Aircraft*, Vol. 31, No. 2, 1994, pp. 288–297.

[8] Sutton, R. S., and Barto, A. G., *Introduction to reinforcement learning*, Vol. 135, MIT press Cambridge, 1998.

[9] van Hasselt, H., and Guez, D., Arthur andvSilver, "Deep Reinforcement Learning with Double Q-Learning," *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, 2016, pp. 2094–2100.

[10] Chollet, F., et al., "Keras," `https://keras.io`, 2015.

[11] MathWorks, "Von Karman Wind Turbulence Model (Continuous)," , 2018. URL `https://mathworks.com/help/aeroblks/vonkarmanwindturbulencemodelcontinuous.html`.