

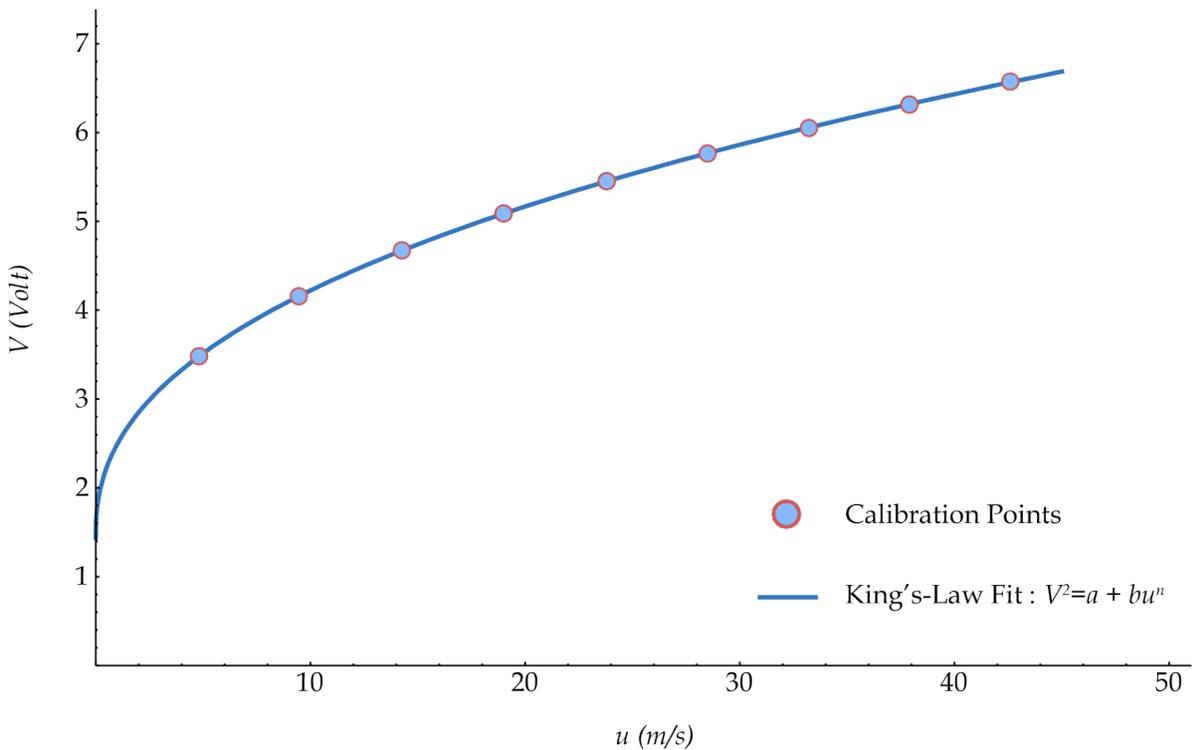
# SM 1: Measurement Setup and Adaptive Control Algorithm

## 1.1 Measurement Setup

We used a hotwire (Dantec 55P15) to measure the streamwise velocity 1 meter downstream of the active grid. The hotwire was operated at constant temperature ( $20^\circ \pm 0.01^\circ$ ) using a CTA-module (Dantec 54T42) and has a maximum sampling frequency of 10 kHz. The hotwire is calibrated using King's Law, which states that

$$V^2 = a + bu^n,$$

where  $V$  is the voltage supplied by the CTA module,  $u$  is the velocity and  $a$ ,  $b$ ,  $n$  are the calibration parameters. We used a set of 9 equidistant velocity calibration points ranging from 0 to 45 m/s. The calibration of the windtunnel is used in order to get a proper calibration of the hotwire. The calibration curve is shown in figure S1.



**Figure S1.** The hotwire is calibrated using King's Law. This curve gives us the velocity as a function of the voltage supplied by the CTA module.

By using the time-resolved velocity signal from the hotwire, the turbulence intensity downstream of the grid is a fluctuating quantity defined as:

$$q(\theta, v, \alpha) = \frac{u_{\text{rms}}}{\bar{u}}$$

where  $u_{\text{rms}}$  and  $\bar{u}$  are the root-mean-square and time-averaged streamwise velocity, given by

$$\bar{u} = \frac{1}{N} \sum_{i=1}^N u_i$$

and

$$u_{\text{rms}} = \sqrt{\frac{1}{N} \sum_{i=1}^N (u_i - \bar{u})^2},$$

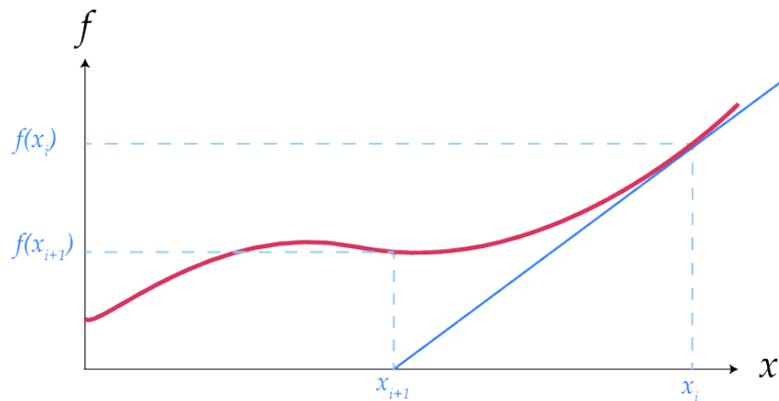
where  $N$  is the sample size and  $u$  is the instantaneous streamwise velocity.

## 1.2 Adaptive Control Algorithm: Newton Raphson Method

The search algorithm is based on the multidimensional variant of the Newton-Raphson method for finding zeroes of an arbitrary function. The Newton-Raphson method is an iterative method based on Taylor series expansion, where each iteration is given by:

$$x_{i+1} = x_i - \alpha \frac{f(x_i)}{\nabla f(x_i)}$$

where  $f$  is the function of which we want to find the zero and  $\alpha$  is a relaxation parameter. The concept behind the algorithm is find the  $x$ -intercept of the tangent line at  $f(x_i)$ , then set  $x_{i+1}$  to this  $x$ -intercept and repeat until  $f(x_i) < \epsilon$ , where  $\epsilon$  is a user-defined tolerance. A schematic representation of this procedure is shown in figure S2.



**Figure S2:** The Newton Raphson method calculates the tangent line at  $x_i$ , then sets  $x_{i+1}$  to the  $x$ -intercept of the tangent line.

In order to use this iterative scheme, we have to assume that our function  $f$  is

differentiable, that is, that  $f \in C^1(\mathcal{R}^n)$ . If this condition holds, we can expand  $f$  in a multidimensional Taylor-series around an arbitrary point  $a \in \mathcal{R}^n$ . For example, in one dimension ( $x$ ), we would write

$$x_{i+1} = x_i - \alpha \frac{f(x_i)}{\frac{df}{dx}(x_i)},$$

where the partial derivative of  $f$  is approximated with a forward-finite-difference scheme using a constant step-size ( $\Delta x$ ):

$$\frac{df}{dx}(a) \approx \frac{f(a+\Delta x) - f(a)}{\Delta x} \equiv \widehat{f}_x(a).$$

To implement the algorithm for our active grid, the function for which we seek zeroes is the error between the actual turbulence intensity  $q$  and the target turbulence intensity  $q_T$ :

$$f(\theta_i, v_i, \alpha_i) = q(\theta_i, v_i, \alpha_i) - q_T$$

We apply Newton-Rhapson in each of the three dimensions of the input space:  $\theta$ ,  $v$ , and  $\alpha$ . We found that a relaxation parameter  $\beta = 1/3$  ensured sufficiently smooth gradients for convergence. The first and  $i$ 'th step, for example, would proceed as follows:

Initial guess:  
 $(\theta_0, v_0, \alpha_0)$

First iteration:

$$\theta_1 = \theta_0 - \frac{1}{3} \frac{f(\theta_0, v_0, \alpha_0)}{\widehat{f}_\theta(\theta_0, v_0, \alpha_0)}$$

$$v_1 = v_0 - \frac{1}{3} \frac{f(\theta_0, v_0, \alpha_0)}{\widehat{f}_v(\theta_0, v_0, \alpha_0)}$$

$$\alpha_1 = \alpha_0 - \frac{1}{3} \frac{f(\theta_0, v_0, \alpha_0)}{\widehat{f}_\alpha(\theta_0, v_0, \alpha_0)}$$

$i$ 'th iteration:

$$\theta_{i+1} = \theta_i - \frac{1}{3} \frac{f(\theta_i, v_i, \alpha_i)}{\widehat{f}_\theta(\theta_i, v_i, \alpha_i)}$$

$$v_{i+1} = v_i - \frac{1}{3} \frac{f(\theta_i, v_i, \alpha_i)}{\widehat{f}_v(\theta_i, v_i, \alpha_i)}$$

$$\alpha_{i+1} = \alpha_i - \frac{1}{3} \frac{f(\theta_i, v_i, \alpha_i)}{\widehat{f}_\alpha(\theta_i, v_i, \alpha_i)}$$

If the value for  $\theta$ ,  $v$ , or  $\alpha$  from the last iteration falls outside the domain of interest, we coerce it to the boundary value of the domain. This coercion is rare, occurring approximately one in every thousand steps, given the small relaxation parameter and the small gradients in the function  $f$ , but we include it here for completeness. Backward-differences are used when  $\theta$ ,  $v$ , or  $\alpha$  are within the upper boundary of the domain of interest within a range of  $\Delta\theta$ ,  $\Delta v$ , or  $\Delta\alpha$ , respectively.

### 1.3 Compact Finite Difference Scheme

Higher order schemes are often used in computational fluid dynamics for obtaining fast convergence. The order of converge of the standard finite difference scheme often increases the number of points on which to sample. Compact finite difference schemes were introduced to circumvent this problem by correlating derivatives and function values at several places. Approximating the gradient in the search space is done by means of a 4<sup>th</sup>-order compact finite difference scheme, given by:

$$\frac{df}{dx}(a + \Delta x) + 4 \frac{df}{dx}(a) + \frac{df}{dx}(a - \Delta x) \approx f(a + \Delta x) - f(a - \Delta x) + O(\Delta x^4).$$

Notice that the scheme requires function evaluations at  $(a + \Delta x)$  and  $(a - \Delta x)$  and not at the center point  $a$ , which makes the scheme ill-suited for the control algorithm. However, the method has a high order of convergence and is well suited for approximating derivatives in spaces with small sets of function evaluations.

## SM 2: Convergence Demonstrated Experimentally for Adaptive Control Algorithm

In order to test the method, we performed 12 sample trials of the adaptive control algorithm in the wind tunnel. For these trials, sampling frequency and sampling time were 10 kHz and 15 s to test the full spectrum case (no down-sampling). The convergence tolerance for turbulence intensity,  $\epsilon$ , was set to 0.005. The results are shown in table S1, where convergence is shown for different initial positions  $(\theta_0, v_0, \alpha_0)$  and target turbulence intensities  $q_T$ .

$\theta_0$ ( $^\circ$ ), $v_0$ (RPM), $\alpha_0$ ( $^\circ$ )	$q_0$	$q_T$	$\theta_{\text{end}}$ ( $^\circ$ ), $v_{\text{end}}$ (RPM), $\alpha_{\text{end}}$ ( $^\circ$ )	$q_{\text{end}}$	# Iterations experiments	#Iterations simulations
8.3, 100, 0	.058	0.1	1.5, 100, 64	0.102	5	4.36±0.83
20.7, 500, 30	.107	0.1	20.7, 500, 29	0.102	1	2.3±1.13
33.2, 900, 60	.339	0.1	18.4, 442, 50	0.099	3	2.86±0.64
8.3, 100, 0	.059	0.2	13.4, 198, 90	0.198	3	3.08±0.92
20.7, 500, 30	.110	0.2	12.2, 1283, 77	0.198	8	9.36±1.22
33.2, 900, 60	.334	0.2	19.4, 467, 53	0.203	6	5.77±0.71
8.3, 100, 0	.058	0.3	33.7, 400, 87	0.301	12	10.01±2.13
20.7, 500, 30	.103	0.3	29.0, 500, 84	0.296	4	3.88±0.40
33.2, 900, 60	.400	0.3	34.3, 1002, 47	0.299	4	4.12±0.79
8.3, 100, 0	.062	0.4	34.3, 871, 81	0.402	12	11.53±0.76
20.7, 500, 30	.906	0.4	30.3, 748, 63	0.396	9	8.71±1.91
33.2, 900, 6	.380	0.4	32.6, 865, 87	0.401	3	1.89±0.81

**Table S1.** The adaptive control algorithm converges in similar number of steps in both experimental and simulated trials. The initial positions ( $\theta_0, v_0, \alpha_0$ ), target turbulence intensity ( $q_T$ ), final experimental conditions ( $\theta_{\text{end}}, v_{\text{end}}, \alpha_{\text{end}}$ ), final experimental turbulence intensity ( $q_{\text{end}}$ ), and number of iterations before convergence are listed for each trial run of the algorithm.