
FINAL YEAR PROJECT

Data-driven estimation and control of flow around two square prisms

Supervisor: Prof. George Papadakis
Second Marker: Dr. Georgios Rigas
Department: Department of Aeronautics
Course: H401 MEng. Aeronautical Engineering
Academic Year: 2022/2023

Student: William Ho
CID: 01717187
Date: 05/06/2023

Abstract

A data-driven method is used to estimate and control flow around two square prisms. The Direct Numerical Simulations (DNS) for the unsteady flow at $Re = 200$, is solved using STAR-CCM+. A reduced order model is obtained using a more efficient Proper Orthogonal Decomposition method via snapshots by extracting the velocity data at certain time instances from the DNS. The **n4sid** subspace identification algorithm is applied to the temporal coefficients to design an optimal linear estimator that approximate the evolution of the temporal coefficients. A model order of 50 with 20 sensors found from the QR pivoting algorithm was successful in reconstructing the far and near wake regions. The reconstruction of small low-energy regions is achieved if more sensors are used. The linear quadratic regulator was designed and combined with the kalman filter to form the LQG controller. A control law to minimise the energy of the temporal coefficients was proposed and therefore only a single parameter, the input cost weighted matrix, needs to be tuned. It was found that $\mathbf{R} = 60\mathbf{I}$ produced favourable results.

Contents

List of Figures	ii
List of Tables	iii
1 Introduction	1
1.1 Objective	2
1.2 Structure	3
2 Acquisition of data: Direct Numerical Simulations	4
2.1 Square prisms	4
2.2 Geometric Setup and Meshing Approach	4
2.3 Solver set-up	5
2.4 Mesh convergence study	6
2.5 Results & Data Extraction	6
3 Reduced Order Model	12
3.1 Mathematical principles	12
3.1.1 POD formulation	12
3.1.2 Snapshot matrix	12
3.1.3 POD eigenmodes	14
3.2 Accuracy of reconstruction	18
4 Estimation	19
4.1 Galerkin projection	19
4.2 Estimator design	19
4.2.1 Subspace System Identification: n4sid	20
4.2.2 Kalman filter	21
4.2.3 QR pivoting algorithm	22
4.3 Design overview	22
4.4 Estimator performance	23
5 Control	28
5.1 Controller design	28
5.1.1 System excitation	28
5.1.2 Linear Quadratic Gaussian	30
5.2 Controller performance	32
6 Conclusions and future work	38
6.1 Future developments	38
A Appendix	41

List of Figures

1	Map of vortex shedding modes shown in [20]	4
2	Geometry and boundary conditions enforced in the setup, image from [5]	5
3	Mesh of the domain and presentation grid in purple	7
4	Fluctuating drag and lift coefficients for the top prism	8
5	PSD spectra of the fluctuating drag (left) and lift (right) coefficients for the top and bottom prisms	8
6	PSD of the fluctuating velocities at the centerline ($y = 0$) with probe points located at $x = 1$ (left), $x = 4$ (middle), $x = 10$ (right)	9
7	FIT [%] of flow at different time instances	10
8	Contours of velocity magnitude of the time-averaged flow	11
9	Streamlines of the time-averaged flow	11
10	Schematic of the snapshot matrix taken from [29]	13
11	The energy distribution of the initial 100 eigenvalues (left) and the cumulative sum (right).	15
12	Temporal coefficient $a_i(t)$ for the first 5 modes	16
13	spatial modes $\phi_i(x)$ of the u' velocity for the first 16 modes	17
14	spatial modes $\phi_i(x)$ of the v' velocity for the first 16 modes	17
15	Reduced order model of the flow domain using 5, 10, 30 and 50 POD modes	18
16	Diagram illustrating the utilization of system identification and Kalman filter in the estimator	23
17	FIT[%] for the first 10 POD temporal coefficients using 20 sensors	23
18	FIT[%] of POD coefficients for different number of sensors	24
19	QR pivoting algorithm vs POD peaks estimation performance	25
20	Reconstructed vs DNS velocity spectra at different probe points	25
21	FIT [%] of flow at different time instances	26
22	Snapshot of the estimation at timestep = 300 for a model order $N_x = 50$, number of modes $m = 50$, and $p = 20$. A video version of this flow simulation can be found online at: https://www.youtube.com/watch?v=YFide5iFmFM	27
23	PSD of excitation signals for the four channels	29
24	Location of actuators	29
25	Schematic of LQG	31
26	RMS velocity of the midline of the flow domain	32
27	v' velocity (top) and v' velocity magnitude signals at $x = 4$	33
28	PSD for DNS and control at $x = 1,4$ and 10	33
29	FIT [%] of flow at different time instances with the controller	34
30	Contour plots of velocity statistics: (top row) $\overline{u'^2}$, (middle row) $\overline{v'^2}$, (bottom row) fluctuating kinetic energy for (left column) DNS and (right column) control, red and blue markers are the u and v velocity sensors, respectively	35
31	Fluctuation of drag coefficient with and without control	36
32	Time average velocity magnitude for DNS and LQG controller	37
33	Signal of the actuation velocity for the 4 actuators	41

List of Tables

1	Mesh independence study	6
2	Aerodynamic coefficients with and without control	36

1 Introduction

The interaction between fluid flows and solid boundaries is found to exhibit complex dynamical patterns. At low Reynolds numbers, the flow separation and regions of highly unsteady recirculation can be influenced by the shape or configuration of the boundary. It is known that for complex flows, the external disturbances are responsible for accelerating the growth of instabilities and changing the evolution of the flow [1]. Especially for bluff bodies, the flow can exhibit formation and shedding of vortices which can cause large, periodic fluctuations in the lift and drag forces acting on the body due to the low pressure vortices shedding from both sides of the body. In real life applications, these vortex shedding can become a major issue as they can cause large oscillations which may cause damage in structures such as high rise buildings. Therefore the study of vortices and its shedding is an important topic of study. However, the investigations of unsteady flows requires the analysis of flow structures at varying times. Therefore, the extraction of accurate models of the flow is important for resolving the spatial and temporal scales. The subject of fluid flow modelling is broad however, the type of modelling for flows with high order dimensionality would usually be resolved with reduced-order modelling techniques, more specifically modal decomposition which Proper Orthogonal Decomposition belongs to [2]. POD works by modelling the flow with spatio-temporal scales by a set of modes derived from a mathematical scheme that involves an extensive amount of linear algebra [3]. The high order dimensionality of the collected data measurements makes the estimation and robust control problematic for the flow models [4]. To mitigate this problem, the use of a low-rank representation of the flow is crucial in the development of data-driven models, in other words a reduced order model as mentioned earlier. Numerous fields, including active flow control, cardiovascular medicine, environmental engineering and more use flow estimation from limited measurements [5].

Fluid flow control is an important discipline in the aeronautics industry. Some applications include improving mixing processes, suppressing instabilities, reducing drag and increasing lift. There are two main methodology to control flow: active and passive control. Passive control aims to control the flow via geometric shaping or the use of passive elements. Examples of passive control used in fluid dynamics applications includes: the control of boundary layer separation through the use of vortex generators and the use of roughness or dimples to control the flow behind a cylinder [6] [7] [8] [9]. The passive controllers have the advantage of being easier to implement due to the absence of moving devices. However, this also means that the passive controller that has been designed cannot be used for different flow configurations. For active control approach, it requires a supply of energy into the system via an actuator. The additional cost of active control is justified by a positive energy balance [10]. Thus if the energy saving of the control system is greater than the energy required for the actuator, the controller is deemed to be acceptable. Examples of actuators used in flow control include plasma devices and small synthetic jets [11]. There are sub categories in active control, namely open-loop control and closed-loop control. In the open-loop scenario, the control law is determined offline. Examples include open loop control of a rotatory movement of a cylinder [12] [13], and control of cavity oscillations with harmonic forcings [14]. Closed-loop control requires measurements of the system in real time to feedback into the system to calculate relevant control actuation. As the measurements are taken in real time, this allows the controller to adapt to any changes in the flow and consequently allows for more efficient and robust control [15]. Even so, the closed-loop control system would now require the computation of both the flow and the actuators which pose as a difficult problem in the applications of fluid flow.

Newer developments have been made to attempt to use genetic algorithms to capture a model-free closed-loop controllers [16]. They work by exploring a wide range of control laws and minimizing a cost function. Controllers obtained from genetic algorithms however have issues with stability robustness and can lead to unstable compensated systems if the conditions are modified [10].

Chaotic flow are fluids that exhibits irregular, unpredictable, and sensitive characteristics. It is characterized by a complex and non-repetitive pattern of fluid motion, where small changes in initial conditions can lead to significant differences in the subsequent flow behavior. The control of these types of flow have been broadly investigated through the use of echo state networks, which is a type of

reservoir computer [17], reinforcement learning (RL) [18] and fuzzy predictive controller [19].

The control of a two-dimensional, low-speed flow past two prisms will be the model chosen due to the unsteady nature of the flow, with its complex vortex shedding features. The attempt to control this type of flow will follow previous numerical and experimental methodology [5] [20]. By adjusting both the Reynolds number and the separation distance between the two prisms, the shedding regimes of this chaotic flow can be identified. The chosen Reynolds number of 200 and separation distance to be equal to the prism's side length was found to produce non-periodic vortex shedding. The applications to real life for these kind of flows can range from blood flows to mixing operations. When it comes to control for chaotic flows, a lot of the methodology is quite complex and relies on non-linear controllers, neural networks, reinforcement learning and more [21]. In the latest experiments for the model with two side by side prisms, a linear estimator was found to be very effective in reconstructing the vortex shedding for this flow. The method involved separating the system identification and the sensor input by including an optimal full state estimator (the Kalman filter). This study also investigated the results of an estimator formulated directed via the data. It was found that the estimator with the kalman filter produced superior results. The study also investigates the problem of sensor placements. A greedy QR pivoting algorithm and sensors placed at POD peaks were investigated and found that the QR algorithm is more accurate in recovering the characteristics of the flow far from the cylinders than sensors placed at the POD peaks. The success of a linear estimator in reconstructing chaotic flow dynamics has prompted the investigation of a linear controller to determine its efficacy. Chaotic flow systems are known for their complex and unpredictable behavior, which poses challenges for control and stabilization. However, linear control techniques offer simplicity and ease of implementation, making them attractive for practical applications. By exploring the potential of a linear controller to effectively regulate chaotic flows, researchers aim to leverage the linear approximation of the system dynamics to achieve stability and desired behavior. This investigation seeks to uncover the extent to which linear control strategies can successfully govern chaotic systems, potentially opening new avenues for control and manipulation of inherently nonlinear and complex flows. A linear quadratic gaussian (LQG) controller has been widely explored within the field of flow control where it has been previously experimented with an open cavity flow and a noise amplifier flow [10] [22]. The outcome of these experiments has proved to be successful when it comes to minimising the perturbation energy for the control case. Again, this will be the aim for the flow mentioned earlier with the use of a linear controller.

1.1 Objective

The main objectives of the project can be shown below:

- Reproduce the chaotic flow of the two square prisms by Direct Numerical Simulations (DNS) that was shown from previous experiments
- Retrieve a reduced order model of the DNS by using a decomposition method attain the spatial and temporal modes of the flow
- Employ the `n4sid` system identification algorithm to extract a state space system
- Use Kalman filter theory to create a linear estimator and use the QR pivoting algorithm to find the optimal locations of the sensors
- Explore the reconstruction performance with different configurations
- Design a linear controller for the chaotic flow that was previously used in [10] and [22]
- Evaluate the performance of the controller on the flow

1.2 Structure

Chapter 1 introduces the reader to the topic of the project and gives a broad range of current studies that have been made.

Chapter 2 will outline the flow physics to be simulated on the relevant computational fluid dynamics (CFD) software and the data extraction method used.

Chapter 3 will cover the Proper Orthogonal Decomposition method that was used in previous studies.

Chapter 4 covers the estimator design, the method used to determine sensor locations and discusses the performance of the estimator

Chapter 5 describes the controller design and assess the performance in supressing the effects of vortex shedding

Chapter 6 summarises the results found from the project and proposes improvements and future work propositions.

2 Acquisition of data: Direct Numerical Simulations

To extract meaningful information of the flow such as coherent structures or patterns, it is important for there to be enough quality data to be extracted. Specifically fluid flow, data would be extracted numerically using simulations or experimentally using techniques such as particle image velocimetry (PIV). A computational approach was taken and CFD simulations were performed on a low-speed, incompressible, 2D unsteady, viscous rotational flow around two side by side square prisms. The Reynolds number of 200 and gap ratio $g^* = \frac{g}{D} = 1$ was implemented like in [5] and [20].

2.1 Square prisms

The study of flow around bluff bodies is crucial in various engineering applications, including enhancing mixing efficiency and implementing flow control techniques. In past studies it was shown that the corner radii can affect the drag [23].

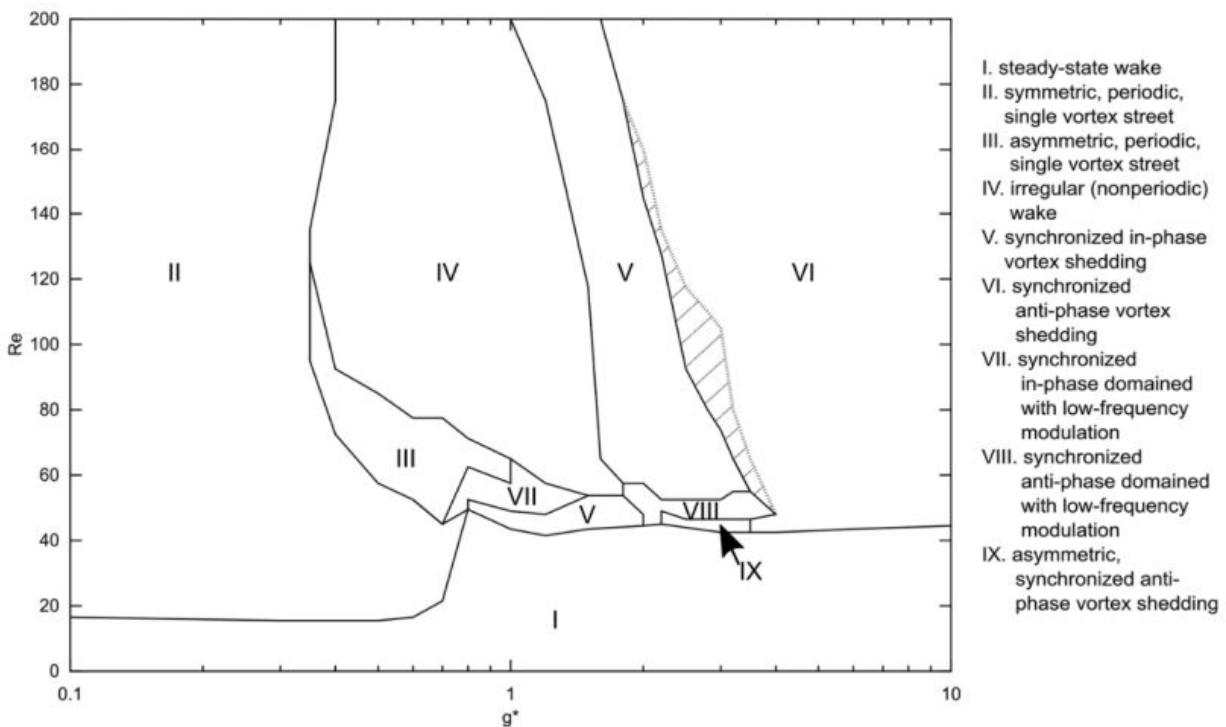


Figure 1: Map of vortex shedding modes shown in [20]

This is because sharp corners promote flow separation. In the present study, the $Re = 200$ and $g^* = 1$ lies within the laminar, irregular, non-periodic region in Figure 1. Given that $Re = 200$ gives laminar flow means that three dimensional effects are small and so the flow behaviour should be realistic. As shown in [5], the selected model configuration would produce a wide range of contents due to the complex and chaotic interactions between vortices.

2.2 Geometric Setup and Meshing Approach

The DNS utilizes a single large rectangular geometry with dimensions of $50D$ in the streamwise direction and $30D$ in the cross-stream direction as seen in [5]. Two simple square prisms side by side with a unit length of $D = 1$, gap length $g = 1$ were removed to represent the solid regions, and hence the gap ratio $g^* = 1$. A smaller rectangular region of horizontal length and vertical length of $14D$ and $8D$, respectively was chosen for the extraction of the velocity data. The bigger region rectangular region was made so that the unsteady wake can develop and no artificial obstruction would occur[24]. The origin was chosen to be the centre between the prisms and centre along the side length of the prisms. There was no particular reason for this choice of coordinate system aside from making the setup easier

to be translated on the chosen CFD software which was STAR-CCM+. The boundary conditions imposed on the domain is listed below:

- The inlet condition is set as a velocity inlet with a constant U_∞ value, chosen as 1 m/s for this study
- The outlet is defined as a pressure outlet so the velocity gradients will be zero (i.e. $\frac{\partial u}{\partial x} = \frac{\partial v}{\partial x} = 0$)
- The top and bottom boundaries are specified as symmetry planes where the u velocity vertical gradient and vertical velocity are zero (i.e. $\frac{\partial u}{\partial x} = v = 0$)
- The no-slip condition $u = v = 0$ is applied to the boundaries on the solid prisms and are defined as walls

The geometric setup and boundary conditions mentioned above can be shown in Figure 2

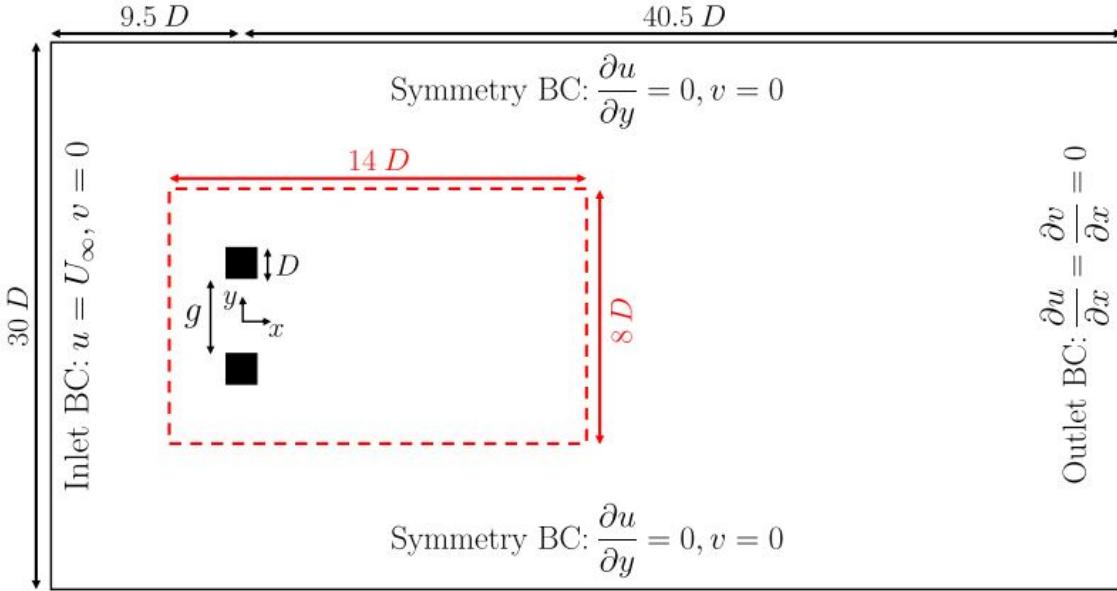


Figure 2: Geometry and boundary conditions enforced in the setup, image from [5]

The flow domain was meshed by ANSYS Mesher using a Cartesian type in previous projects. For STAR-CCM+, to imitate this meshing type, a directed structured mesh was used following the tutorial in [25]. The reason for this type of mesh is to retain important data at a high level of detail such as the wake eddies, the flow between the gap and close to the prisms whilst having data that will not be extracted such as the far regions to be coarse to save computational time.

2.3 Solver set-up

As previously mentioned, we are examining the flow of an incompressible, two-dimensional fluid around two square prisms with side length D . The prisms are positioned apart from each other by a gap of size g . The configurations of such flow can be modelled by the Navier-Stokes equation. The flow is governed by the following continuity and momentum equations:

$$\nabla \cdot \mathbf{u} = 0 \quad (1a)$$

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p + Re^{-1} \nabla^2 \mathbf{u} \quad (1b)$$

Where \mathbf{u} is the non-dimensional velocity vector and p is the pressure. The reference quantity for

distances is D , for velocity it is the flow speed U_∞ and for pressures, it is ρU_∞^2 where ρ is the density of the fluid [5]. The Reynolds number Re is defined as $Re = U_\infty \frac{D}{\nu}$ where ν is the kinematic viscosity. Additionally, for any general variable ϕ , it can be composed of the time average of the general variable and the fluctuation of the general variable. More explicitly shown in equation 2

$$\phi = \bar{\phi} + \phi' \quad (2)$$

Where $\bar{\phi}$ and ϕ' are the mean and the fluctuation of the general variable, respectively. The model was computationally solved using the finite volume method with a second-order accuracy, employing cell centers as the computational points. A 2nd order accurate implicit PISO solver was used to enforce the incompressibility condition and additionally it does not require internal iterations[26][27]. Another parameter to consider is the Courant-Friedrichs-Lowy (CFL) number which is recommended to be smaller than 1 to ensure stability. However, the solver used is implicit and therefore the CFL number will only aid in the accuracy of the mesh convergence study. A time step of 0.01 was chosen as it meets the requirement of the CFL number.

2.4 Mesh convergence study

The validation of the discretisation of the DNS was compared to numerical results found in [5] [20]. The mesh convergence study explored a wide range of mesh sizes, however only three different sizes will be shown. The drag coefficient can be monitored on STAR-CCM+ and exported in a CSV file where the mean is taken. The strouhal number was calculated which can be obtained by finding the frequency between peaks [26].

Table 1: Mesh independence study

N. cells	Side length subdivision	Δt	\bar{C}_D	\bar{C}_L	St
55500	20	0.01	2.054	0.218	0.167
98000	30	0.01	2.035	0.204	0.168
135500	50	0.005	2.033	0.201	0.168

From table 1, it is evident to see that as the number of global cells increase, the aerodynamic coefficients converges closer to the values found in [20] (\bar{C}_D and \bar{C}_L are found to be 2.03 and 0.19, respectively). The change in values for the middle and finest mesh is seen to be marginal and can be said to have converged. The middle mesh was chosen over the more accurate finest mesh due to the minimal benefit it produces. Given that the simulation will require a long runtime for the statistics to have converged, the computational cost was taken into consideration, and a slightly coarser mesh will remedy this situation.

2.5 Results & Data Extraction

The simulation was carried out for 1000 time units where the first 100 time units were truncated and only data from the following 900 time units were extracted to ensure the statistics of data has converged. The extraction rate $\Delta t_{extract}$ was kept to be the same as the DNS timestep (i.e. $\Delta t_{extract} = \Delta t$) so there would be 100 velocity snapshot per time unit. This results in 90,000 snapshots (csv files) collected for further analysis with 11,259 spatial points. The extraction of velocity snapshots at a desired timestep can be performed with the use of XYZ tables on STAR-CCM+. The region outlined for extraction shown by the dotted red lines in Figure 2 can be created in STAR-CCM+ using a type of probe called a 'presentation grid'. An image of the presentation grid used on STAR-CCM+ is shown in Figure 3.

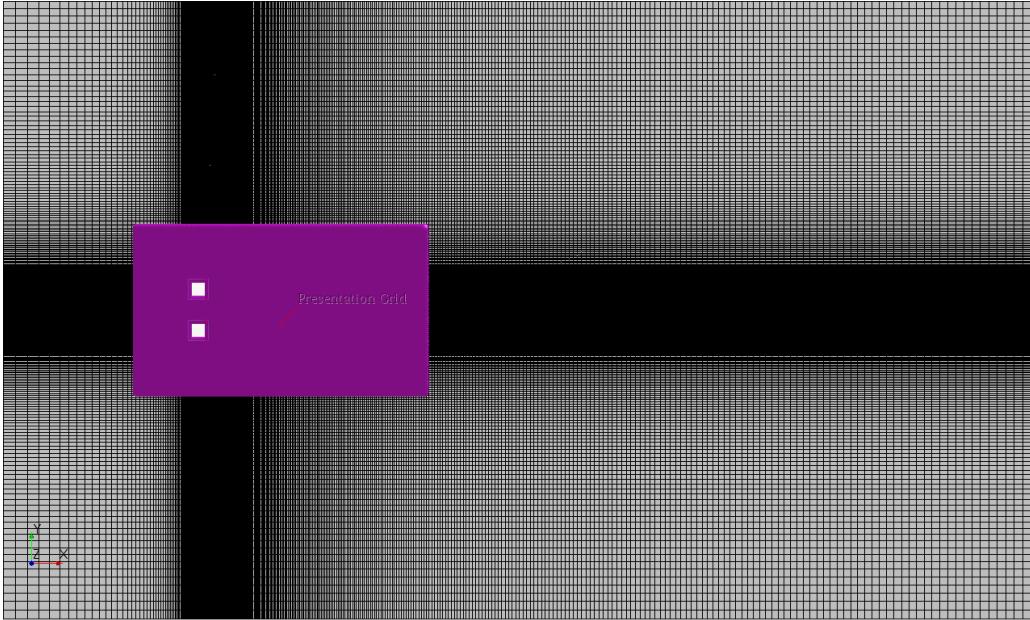


Figure 3: Mesh of the domain and presentation grid in purple

STAR-CCM+ provides an option to use 'reports' to record and create plots of the aerodynamics coefficients against time. The aerodynamic coefficients' time histories can be shown on Figure 4, where it displays chaotic behaviour. It will be shown later how and if the fluctuations can be damped using a linear controller. It must be noted that the signals are centered around their means and the underlying structure of the flow is hard to analyse with this plot itself.

The flow properties and characteristics can be understood better through a visualisation of the flow field in the frequency domain by calculating the Power Spectral Density (PSD) of these fluctuations. The PSD for the drag and lift coefficients is plotted against the Strouhal number, displayed on Figure 5 for both the top and bottom prisms. It can be seen from the PSD plot that there is good overlapping between the top and bottom prisms which suggests the time average flow is almost symmetric. Additionally, there is clearly a prominent peak in both the drag and lift spectra, namely at $St = 0.161$. In addition, the drag coefficient displays rich signals at lower frequencies such as $St = 0.013, 0.054$ and 0.08 . It should be noted that the integration time was 1,000 time units, which is over half the integration time used in previous projects, and therefore the slowest temporal scale ($St = 0.013$) can resolve around 13 cycles. This is a lot less in comparison to [5] which resolved over 50 cycles. However, it can be seen later that there is no large discrepancy in results as these observations align with what is seen in [20]. The spectra indicates that this particular flow is a high-order dynamical system since there are multiple dominant characteristic shedding frequency.

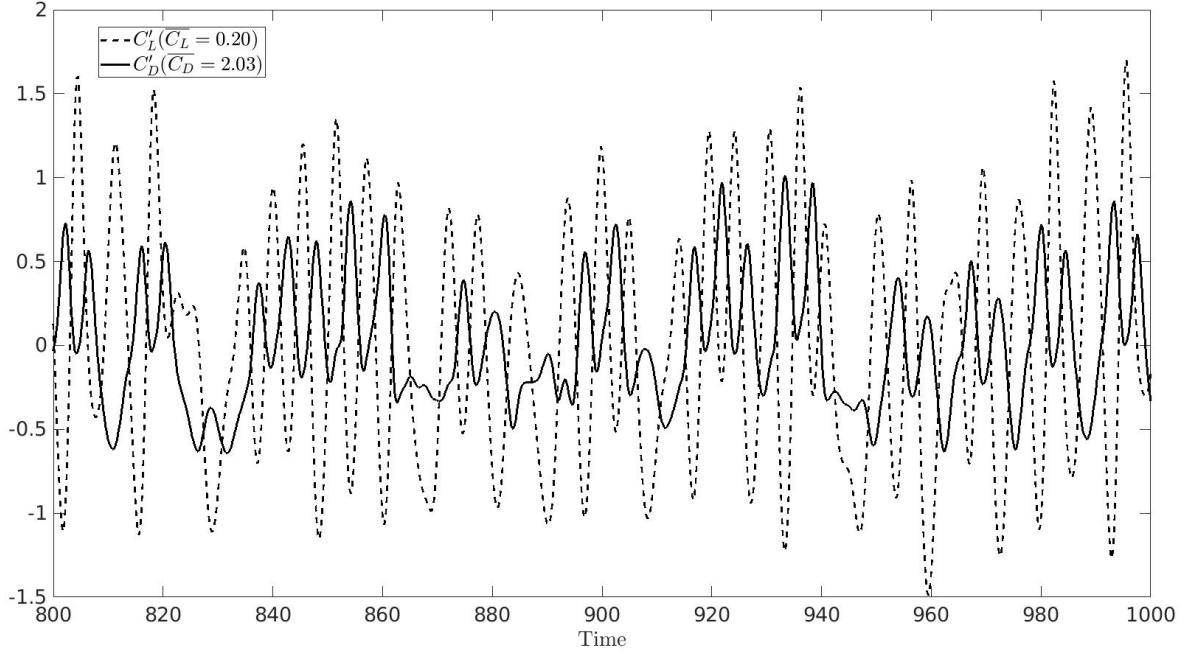


Figure 4: Fluctuating drag and lift coefficients for the top prism

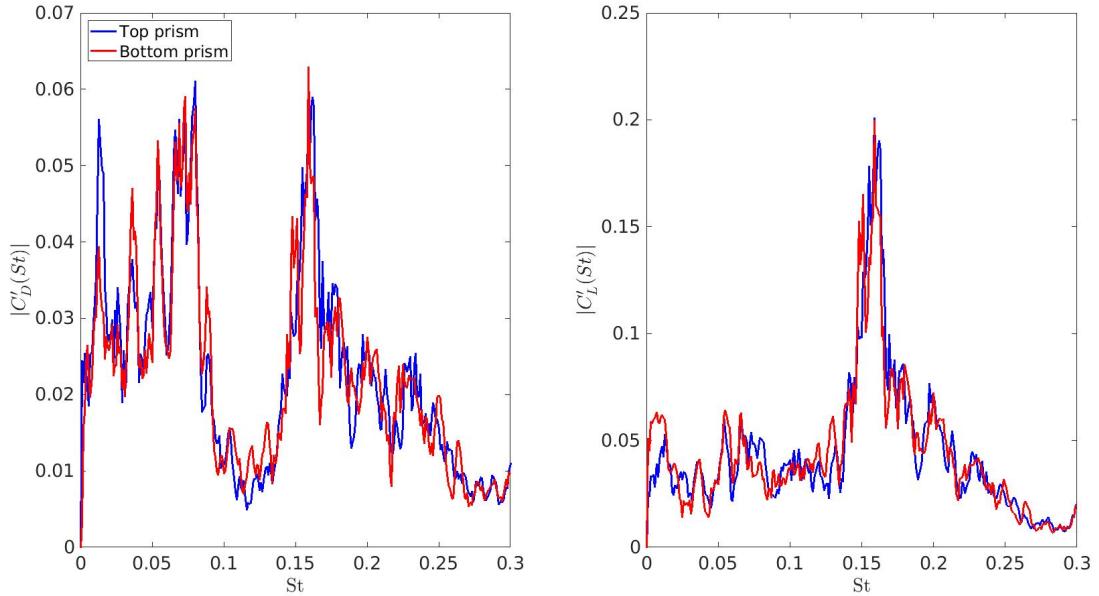


Figure 5: PSD spectra of the fluctuating drag (left) and lift (right) coefficients for the top and bottom prisms

From literature, the Strouhal number for a flow past a single square cylinder at $\text{Re} = 200$ was recorded to be 0.15 [20][24]. For the present case, a large proportion of the energy content has shifted towards $\text{St} = 0.161$. The interaction between two prisms have also initiated fluctuations at slower temporal scales. As explained in [5] and analysed by [20], the erratic motion of the jet originating from the gap between the prisms contributes to the flow development downstream of the wake. Depending on the gap flow orientation rear of the prisms, it displays varying size of the upper and lower wakes. Although the unsteady flapping motion from the gap is seen as non-periodic, over a large period of time, orientation of the flapping jet will have equal probability and so the time average flow will be symmetric.

The effect of the flapping flow is analysed by recording the velocity fluctuations at different probe points on the center line. The PSD of the different probe points can be seen on figure 6. Close to the rear of the prisms, the spectra can be seen on the left where it displays the u' and v' in blue and red, respectively. Overall, the spectra pattern aligns with what is seen in [5]. It can be seen that close to the gap, a $St = 0.147$ is pronounced for both velocities. The v' also exhibits a large proportion of energy at $St = 0.078$. This was explained in [5] that the vortex shedding affects both the velocities but the flapping motion only affects the v' velocity. This would suggest that the flapping motion contributes to the lower frequency region. At $x = 4$, the v' spectra shows dampening of the very small frequencies (i.e. $St \approx 0.046$), which would be understandable since the probe would detect less impact from the flapping jet as it is further away. Additionally, the peak of 0.147 has become slightly more pronounced at $x = 4$, compared to $x = 1$. Moreover, the energy content for the u' has increased for the low frequency region which was suggested to be because of the sub-harmonic of the v' at $x = 1$ at the lower frequency region[5]. The probe at $x = 10$ shows the shedding frequency of $St = 0.147$ being damped for both velocities. These images clearly depicts the effects of the flapping jet from the gap. As the flow goes from left to right, the system exhibits two characteristic frequencies to a wider range of small and intermediate frequencies.

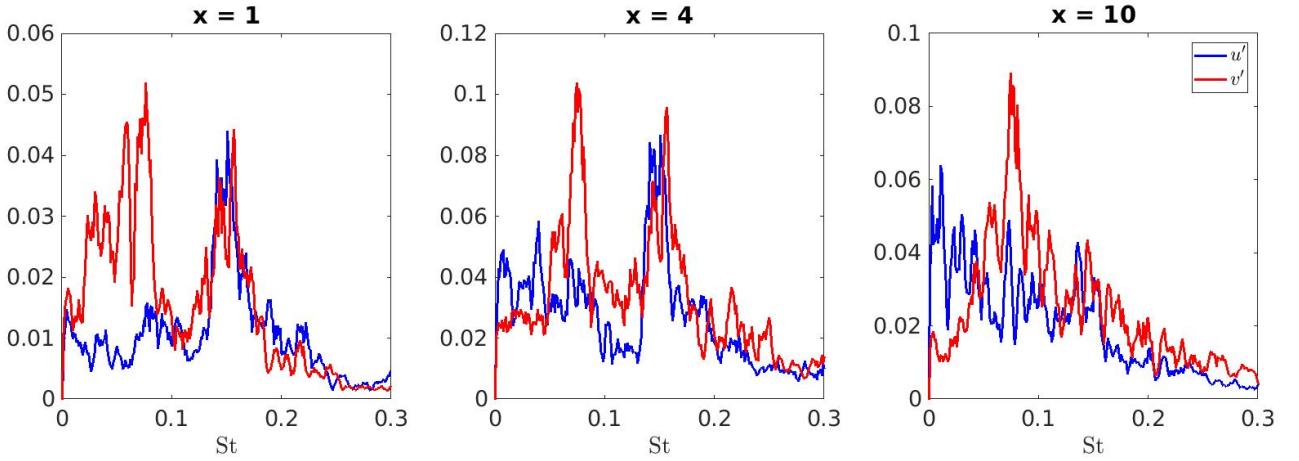


Figure 6: PSD of the fluctuating velocities at the centerline ($y = 0$) with probe points located at $x = 1$ (left), $x = 4$ (middle), $x = 10$ (right)

Despite the flow displaying unpredictable and chaotic dynamics, the time-averaged flow demonstrates symmetry or close to symmetry. This can be seen with the velocity magnitude contours and the streamline of the time average flow in figure 8 and 9.

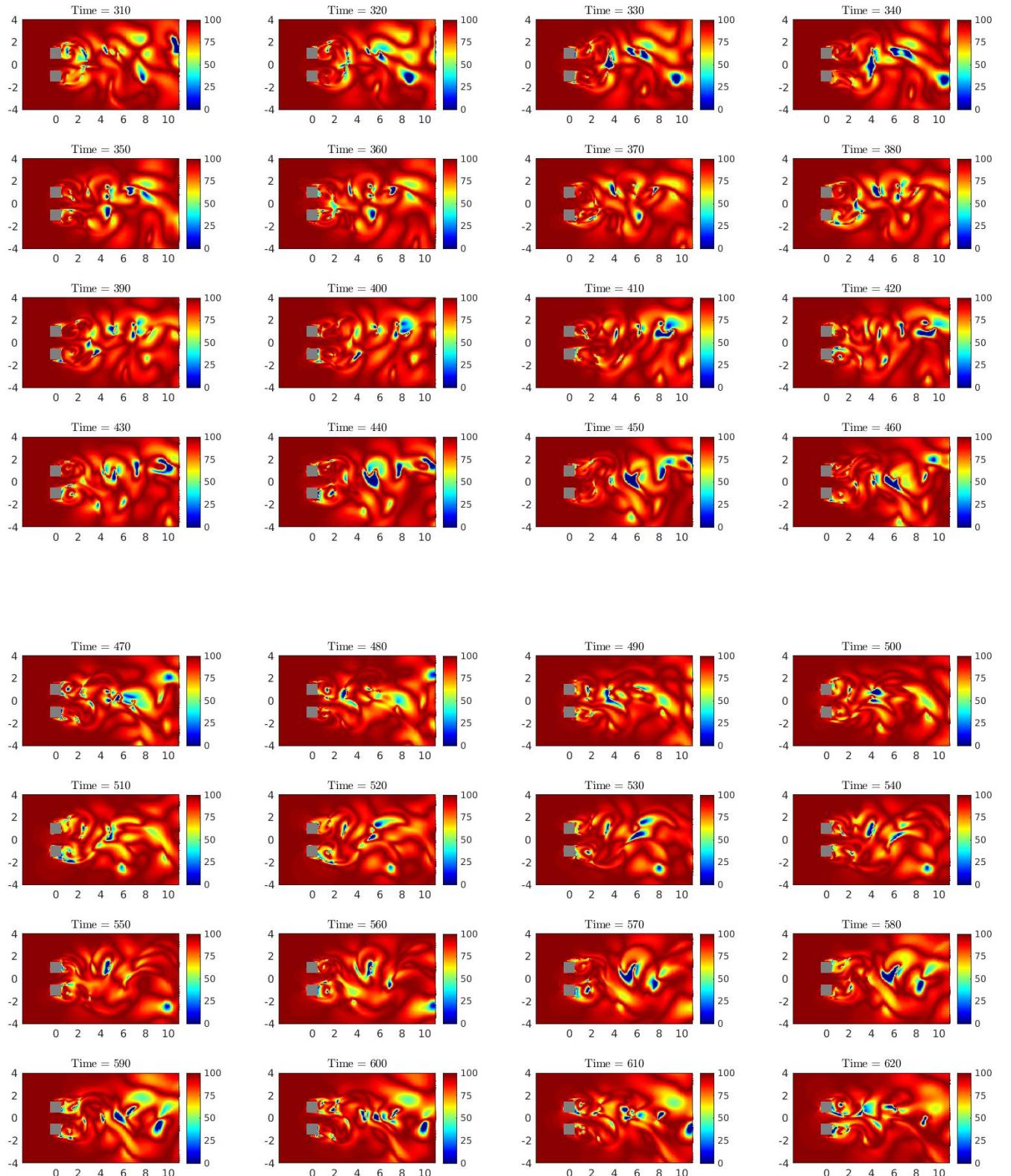


Figure 7: FIT [%] of flow at different time instances

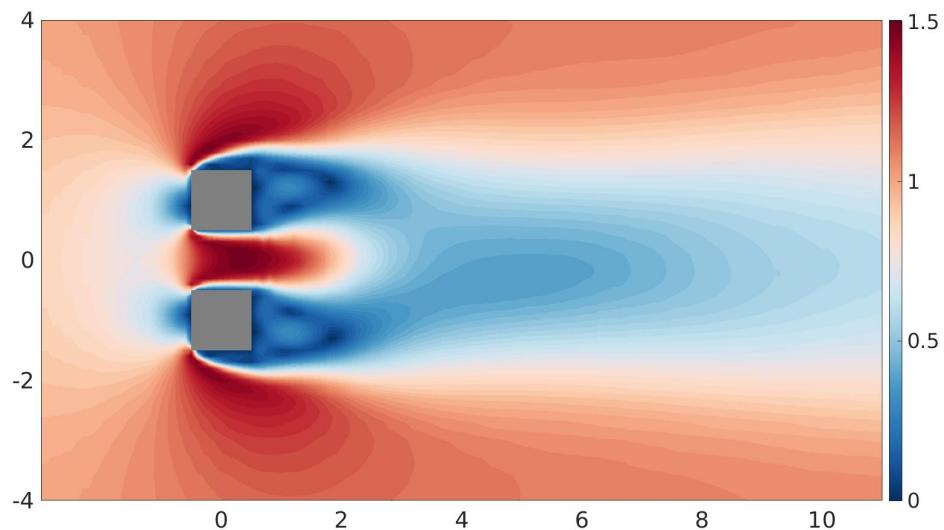


Figure 8: Contours of velocity magnitude of the time-averaged flow

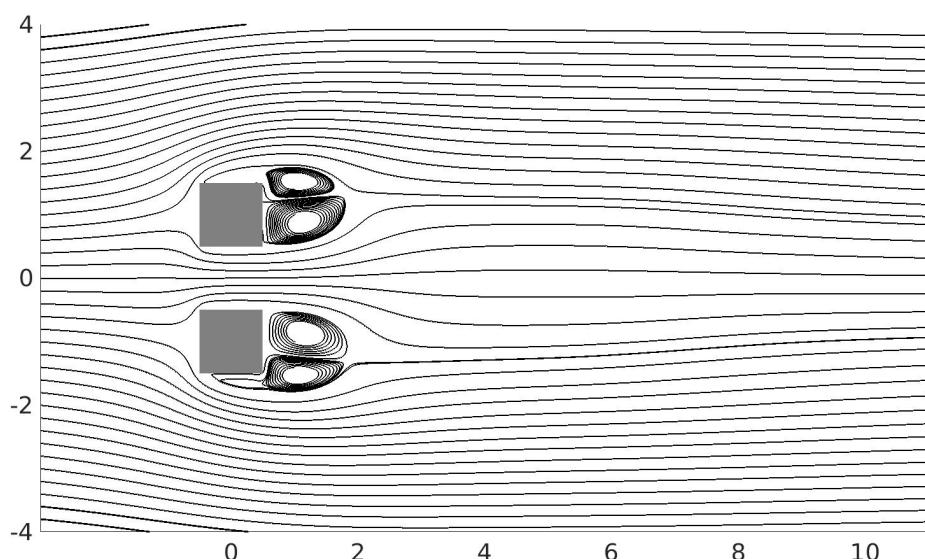


Figure 9: Streamlines of the time-averaged flow

3 Reduced Order Model

Proper Orthogonal Decomposition (POD) is a mathematical technique used to extract dominant spatial and temporal modes from data sets and was first introduced to the fluid mechanics research community by Lumley in 1967 [2]. It is extensively applied in the field of data analysis, fluid dynamics, and structural mechanics. In other fields, it is also known as Principle Component Analysis (PCA) in statistics, Singular Value Decomposition (SVD) in linear algebra and Karhunen-Loëve decomposition. The method aims to identify the most significant patterns or structures within a given dataset by decomposing it into orthogonal modes. These modes capture the major variations and dominant features present in the data. By reducing the dimensionality of the dataset, POD allows for efficient representation and analysis of complex systems. The POD method can be presented as part of a broader family of modal decomposition methods, such as Balanced POD and Koopman analysis. It has proven to be a valuable tool in understanding and modeling physical phenomena, enabling insights into underlying dynamics, system behavior, and optimization of various processes. This method is not only used for the application of fluid flow however, as it is commonly employed in data compression, image processing and system identification.

This chapter will include an overview of how the POD method is orchestrated and show the comparison between the original DNS model and the reduced-order model (ROM).

3.1 Mathematical principles

This section will detail the steps behind the POD method. Additionally, the Method of Snapshots introduced by Sirovich [28] was presented due to the superior computational efficiency in comparison to the standard POD method. More detail can be found in [2].

3.1.1 POD formulation

As expressed in equation 2, the quantities obtained from the DNS can be separated into fluctuating and time averaged mean terms via Reynolds decomposition. This will be extended to such that the equation holds for various spatial coordinates and time instances. The modification can be shown below:

$$\mathbf{u}(\mathbf{x}, t) = \bar{\mathbf{u}}(\mathbf{x}) + \mathbf{u}'(\mathbf{x}, t) \quad (3)$$

where $\bar{\mathbf{u}}$ represents the time averaged flow field and independent of t [5]. It is now possible to decompose the fluctuation of velocity into a linear combination of spatial (ϕ) and temporal coefficients (a). The velocity fluctuations can be expressed as:

$$u'(x, t) = \sum_{i=1}^{\infty} \phi_i(x) a_i(t) \quad (4)$$

The POD works by finding a set of *optimal* basis $\phi_i(\mathbf{x})$ that maximises the variance in the fluctuations field contained within the first $m \ll \infty$ modes [5].

3.1.2 Snapshot matrix

As mentioned earlier, the snapshots method was first introduced by Sirovich [28]. We can form the snapshot matrix by letting the velocity measurements of a single time step be defined in a column vector (i.e. a snapshot of the velocity fluctuation for a given time will be represented as a column vector containing all spatial locations), with all other time steps being concatenated along the next column. The snapshot matrix of the velocity fluctuations can be expressed as $\mathbf{Y}(\mathbf{x}, t)$ as shown below:

$$\mathbf{Y}(\mathbf{x}, t) = \begin{bmatrix} u'^{(1)}(\mathbf{x}_1) & u'^{(2)}(\mathbf{x}_1) & \dots & u'^{(M)}(\mathbf{x}_1) \\ \vdots & \vdots & \dots & \vdots \\ u'^{(1)}(\mathbf{x}_n) & u'^{(2)}(\mathbf{x}_n) & \dots & u'^{(M)}(\mathbf{x}_n) \\ v'^{(1)}(\mathbf{x}_1) & v'^{(2)}(\mathbf{x}_1) & \dots & v'^{(M)}(\mathbf{x}_1) \\ \vdots & \vdots & \dots & \vdots \\ v'^{(1)}(\mathbf{x}_n) & v'^{(2)}(\mathbf{x}_n) & \dots & v'^{(M)}(\mathbf{x}_n) \end{bmatrix} \in \mathbb{R}^{2n \times M} \quad (5)$$

where \mathbf{x} as before represents the spatial coordinates (i.e. $\mathbf{x} = [x, y]$ for the two dimensional case) and the total number of spatial coordinates and the number of extracted snapshots is represented by n and M , respectively. The schematic of what the snapshot matrix would look like physically is given in figure 10.

For the DNS described in the previous chapter, n and M are 11,259 and 90,000, respectively. It should be noted that equation 5 can be extended for higher dimensions. For example, if an experiment was simulated with a three dimensional model, the velocity in the third coordinate direction (w) can be stacked below the matrix shown above and form a matrix $\mathbf{Y}(\mathbf{x}, t)$ with dimensions $3n \times M$, belonging to the set of real numbers, denoted as $\mathbf{Y} \in \mathbb{R}^{3n \times M}$.

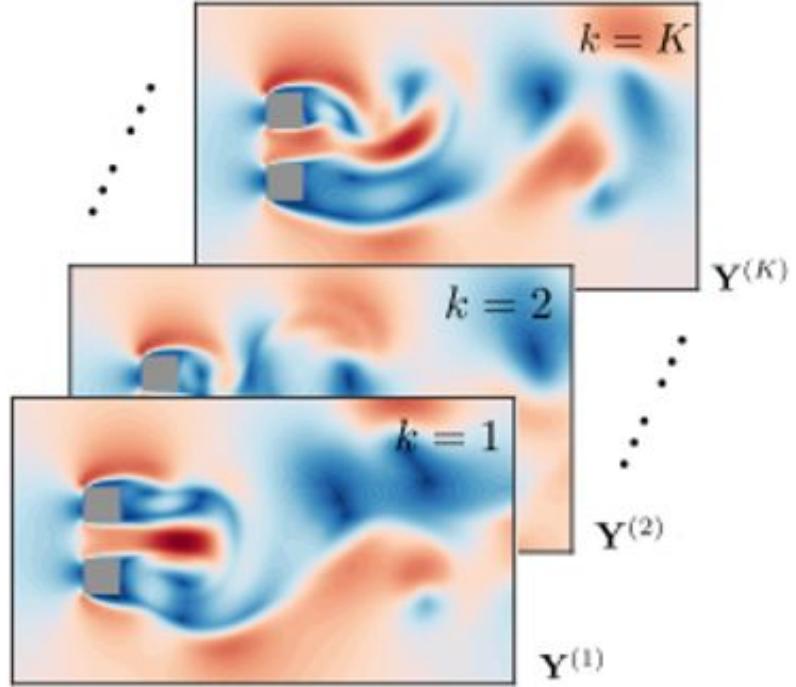


Figure 10: Schematic of the snapshot matrix taken from [29]

Holmes *et al* [30] demonstrates that the POD modes can be found from the singular value decomposition of the $\mathcal{V}^{1/2}\mathbf{Y}$ matrix, where \mathcal{V} is the mesh topology matrix that contains the area of each cell ($\Delta x \Delta y$) along the leading diagonal [5].

$$\mathcal{V} = (\Delta x \Delta y) \mathbf{I}_{2n \times 2n} \quad (6)$$

$\mathbf{I}_{2n \times 2n}$ in equation 6 is the identity matrix of size $2n \times 2n$. The singular value decomposition for a uniform mesh can be described as:

$$\mathcal{V}_{1/2}\mathbf{Y} = \mathbf{U}\Sigma\mathbf{V}^T \quad (7)$$

where $\mathbf{U} \in \mathbb{R}^{2n \times M}$ is the matrix of left singular vectors whose columns contains the unscaled spatial coefficients also known as POD modes [5]. $\Sigma \in \mathbb{R}^{M \times M}$ is the matrix of singular values arranged in descending order along the diagonal as the importance of the modes decreases, and $\mathbf{V} \in \mathbb{R}^{M \times M}$ is the matrix of right singular vectors. Note that \mathbf{U} and \mathbf{V} are orthogonal matrices, i.e. $\mathbf{U}^T\mathbf{U} = \mathbf{U}\mathbf{U}^T = \mathbf{I}$ meaning that $\phi_i \cdot \phi_j = \delta_{ij}$ is the Kronecker delta [26]. To aid faster computation, MATLAB offers a setting called 'econ' to be enabled where it removes the rows and columns with zero entries in Σ and the corresponding \mathbf{U} and \mathbf{V} columns [31].

The scaled POD modes are the columns of

$$\phi(\mathbf{x}) = \mathcal{V}^{-1/2}\mathbf{U} \quad (8)$$

The relationship between the eigenvalues and the corresponding singular values ϕ_i is expressed as:

$$\lambda_i = \frac{\phi_i^2}{M} \quad (9)$$

The snapshots can be projected onto the POD spatial modes to compute the temporal coefficients:

$$\mathbf{a}(t) = \mathbf{Y}^T\mathcal{V}^{1/2}\mathbf{U} \quad (10)$$

where $\mathbf{a} \in \mathbb{R}^{M \times M}$ is a matrix of temporal coefficients ($a_i(t)$).

Now all is left is to reconstruct the original flow by retaining a handful of modes and truncating the rest. The number of POD modes required to accurately reconstruct the flow is dependent on the Reynolds number and the complexity of the flow. The choice of the number of modes, m can be chosen by visualising the cumulative energy content with each additional POD mode. This can be expressed mathematically where we have an approximation of equation 4 given as

$$\mathbf{u}'(\mathbf{x}, t) \approx \sum_{i=1}^{m \ll M} \phi_i(\mathbf{x}) a_i(t) \quad (11)$$

3.1.3 POD eigenmodes

The purpose of projecting the spatial flow structure onto orthogonal modes is to facilitate an understanding of vortex interactions within the prisms. The normalised eigenvalues and their cumulative sum are plotted in figure 11. The figure shows how much energy the modes contribute to the system as the modes gets higher. The first six modes shows that it contains roughly 62% of the fluctuation velocity energy. For a choice of 50 modes, it is able to capture over 97% of the energy content. It can be seen that for higher modes, while technically part of the system, have minimal impact on the overall behavior and dynamics. Therefore, pursuing the inclusion of these additional modes would introduce unnecessary complexity without significant benefits. Considering the computational power required to analyze and simulate these extra modes, it becomes evident that the cost outweighs the potential gains. Thus, it is not deemed worthwhile to incorporate these additional modes into the system, as their contribution to the energy and overall behavior is minimal.

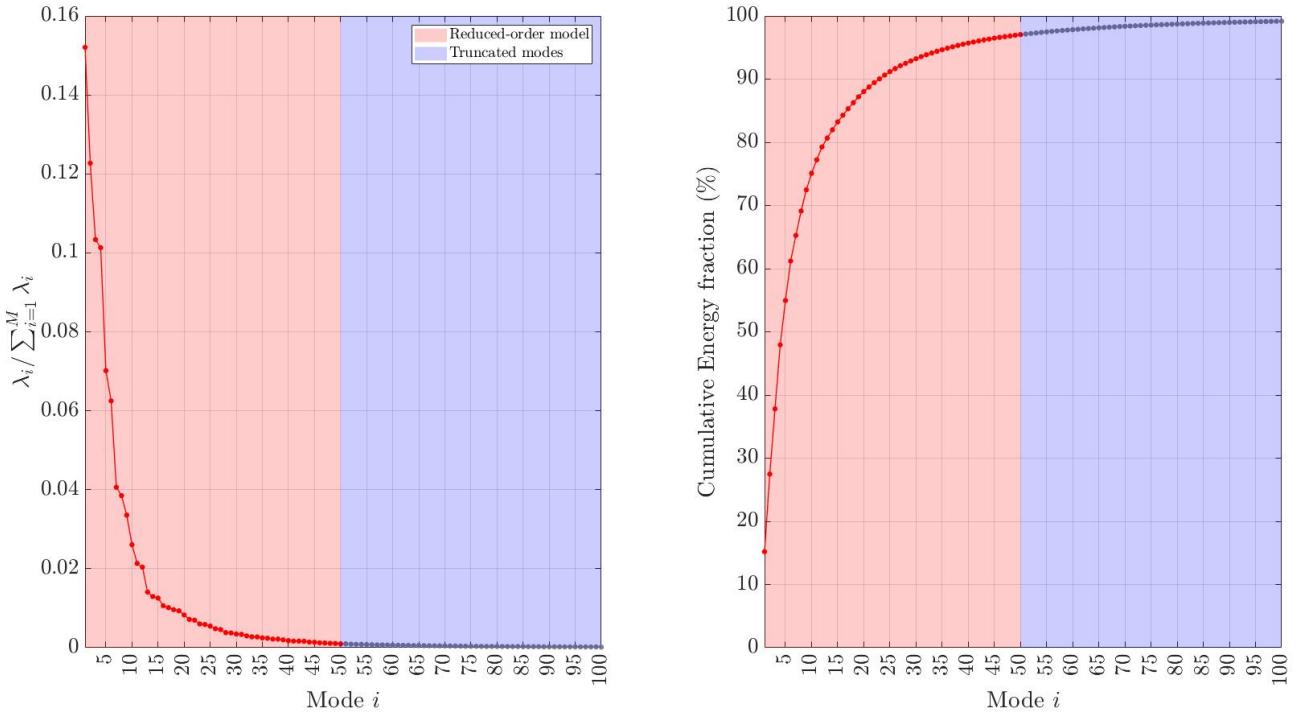


Figure 11: The energy distribution of the initial 100 eigenvalues (left) and the cumulative sum (right).

An observation can be made from the left plot on figure 11 that the POD modes do not appear to be strong pairs, or have exactly the same energy that would normally occur under regular vortex shedding [32]. There are however some modes with very similar energy content that could be due to the irregular vortex shedding pattern.

It can be seen on figure 12, that for the first four modes, the pair of modes exhibit the same frequency peak. It can be seen that for the first pair of modes, the peak frequency is at $St = 0.08$, which belongs to the flapping jet. The next pair of modes have a peak frequency of $St = 0.15$, which stems from the main vortex shedding on the prisms [5].

The first pair of spatial modes also shows an anti-symmetric pattern for u' but a symmetric pattern for v' along the centre line. The next pair of spatial modes then express the complete opposite behaviour where now the u' is symmetric and the v' is anti-symmetric along the centre line.

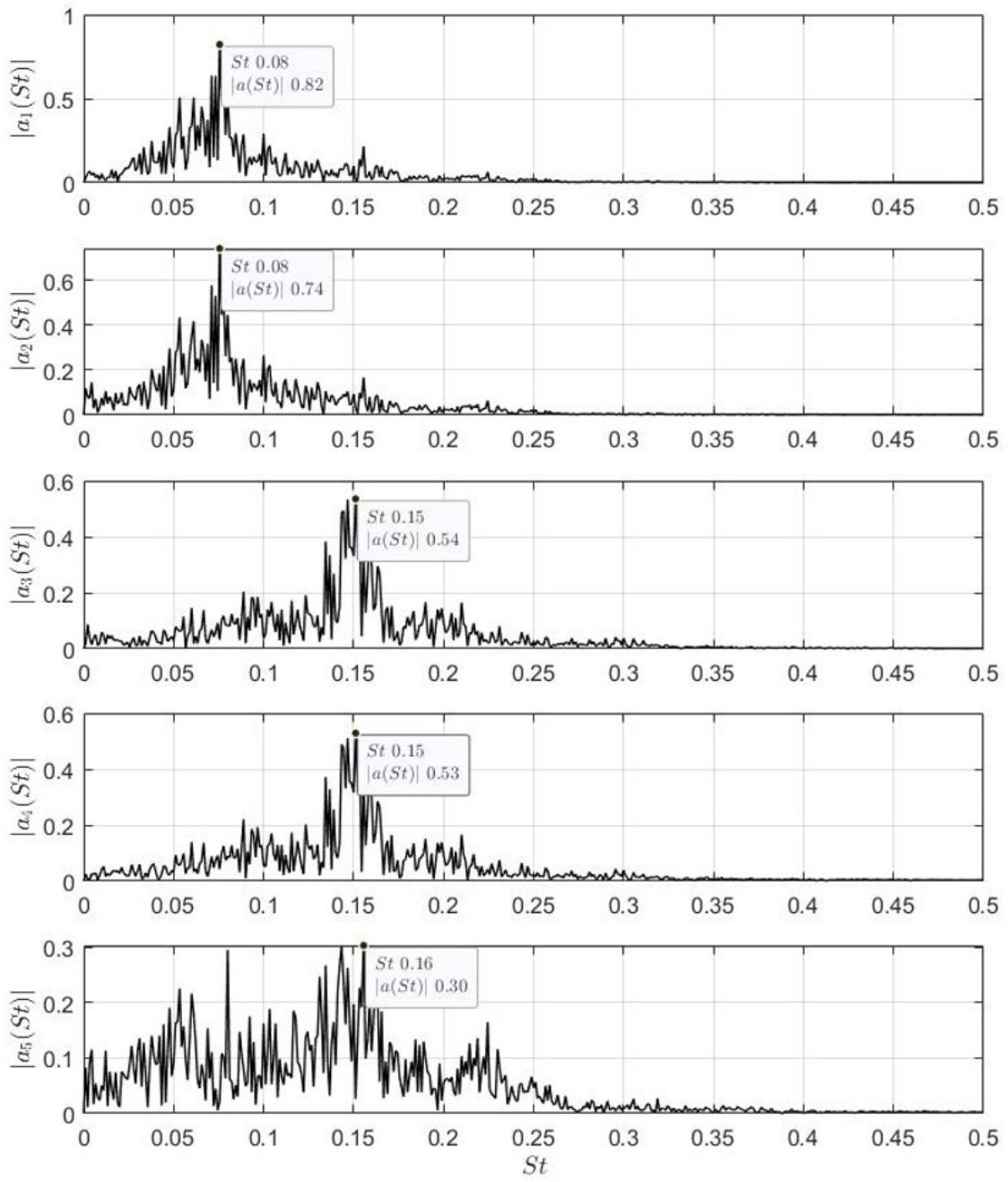


Figure 12: Temporal coefficient $a_i(t)$ for the first 5 modes

The pairs of mode analysed, would suggest that the each of them correspond to a spatial shift (i.e. mode two is a spatial shift of mode 1 and etc), more noticeable on figure 14.

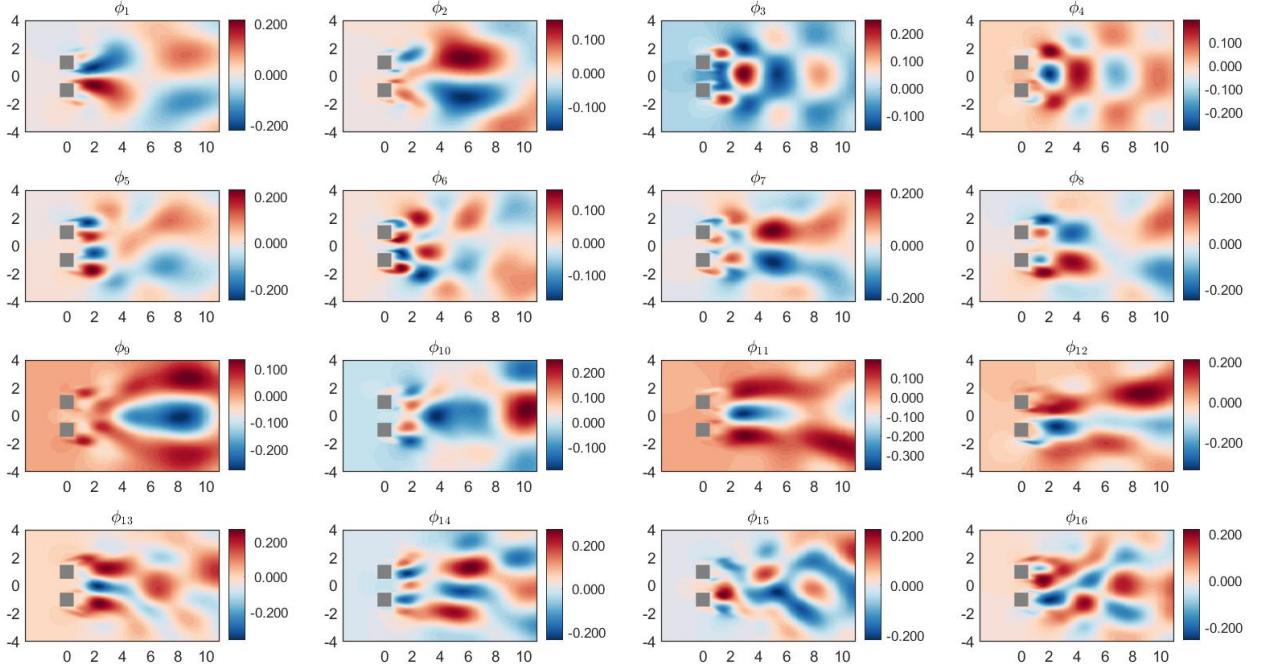


Figure 13: spatial modes $\phi_i(x)$ of the u' velocity for the first 16 modes

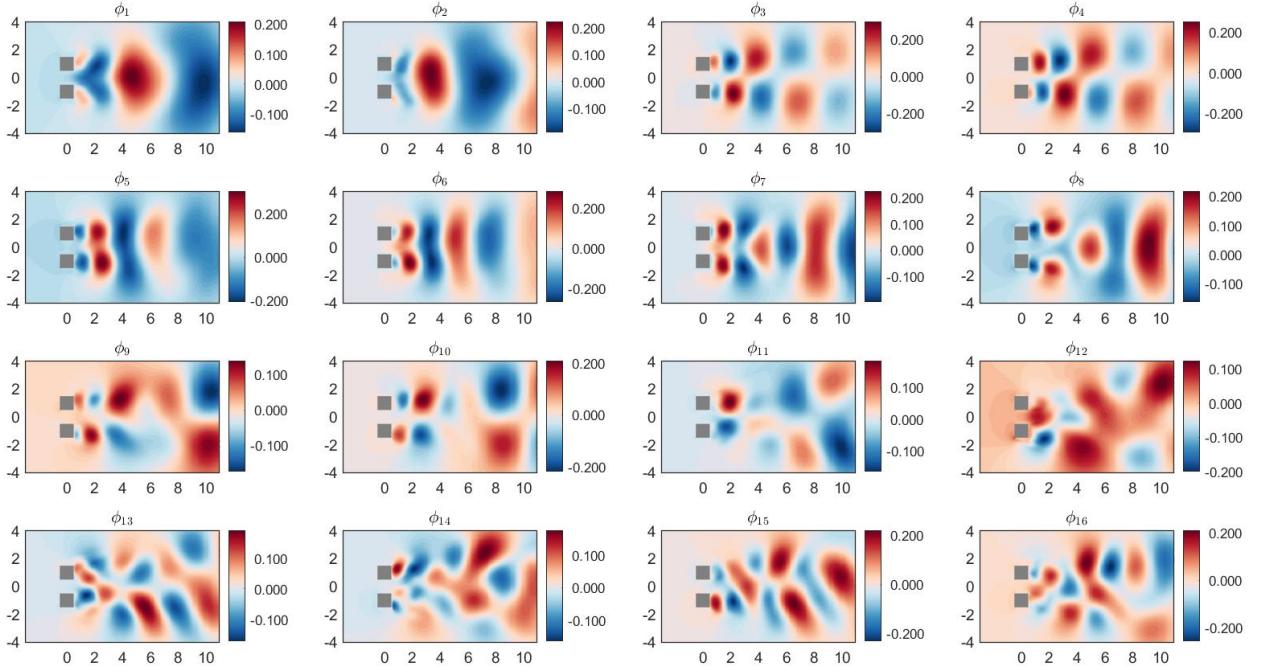


Figure 14: spatial modes $\phi_i(x)$ of the v' velocity for the first 16 modes

For higher spatial modes, the structure of the flow seems to be more unorganised and therefore more difficult to interpret. This would seem sensible as more noise is in the modes and interpretability becomes lost.

3.2 Accuracy of reconstruction

Flow reconstruction is the process of multiplying the spatial modes with the corresponding temporal coefficients. The reconstruction of the velocity flow field can be obtained by adding the reduced order snapshot matrix with the time averaged mean velocity. The accuracy attained for different ROM orders in reconstructing the velocity magnitude for a certain time instant can be found by finding the FIT [%]. The formula for the FIT is expressed in equation 12:

$$FIT[\%] = 100 \left(1 - \left| \frac{X_{reconstruct} - X}{X} \right| \right) \quad (12)$$

where $X_{reconstruct}$ is the reconstructed version of the variable X . In this scenario, X is defined as the magnitude of the velocity, $X = \|\mathbf{u}(\mathbf{x}, t)\| = \sqrt{u^2(\mathbf{x}, t) + v^2(\mathbf{x}, t)}$. The snapshot chosen was at $t = 200$. The reconstruction on figure 15 shows the DNS of the velocity magnitude result on the left hand side, the reconstruction result in the centre and the FIT of the flow domain. For modes 5-10, the reconstruction is unable to accurately depict the vortex shedding region and the wake. This issue is nullified as more modes are added. It is evident that as more modes are added the reconstruction improves as the amount of blue region of the FIT figures decreases. Although 50 modes may seem to be a relatively large reduced order model, this model order is much lower in comparison to the maximum model order which is $M = 90,000$.

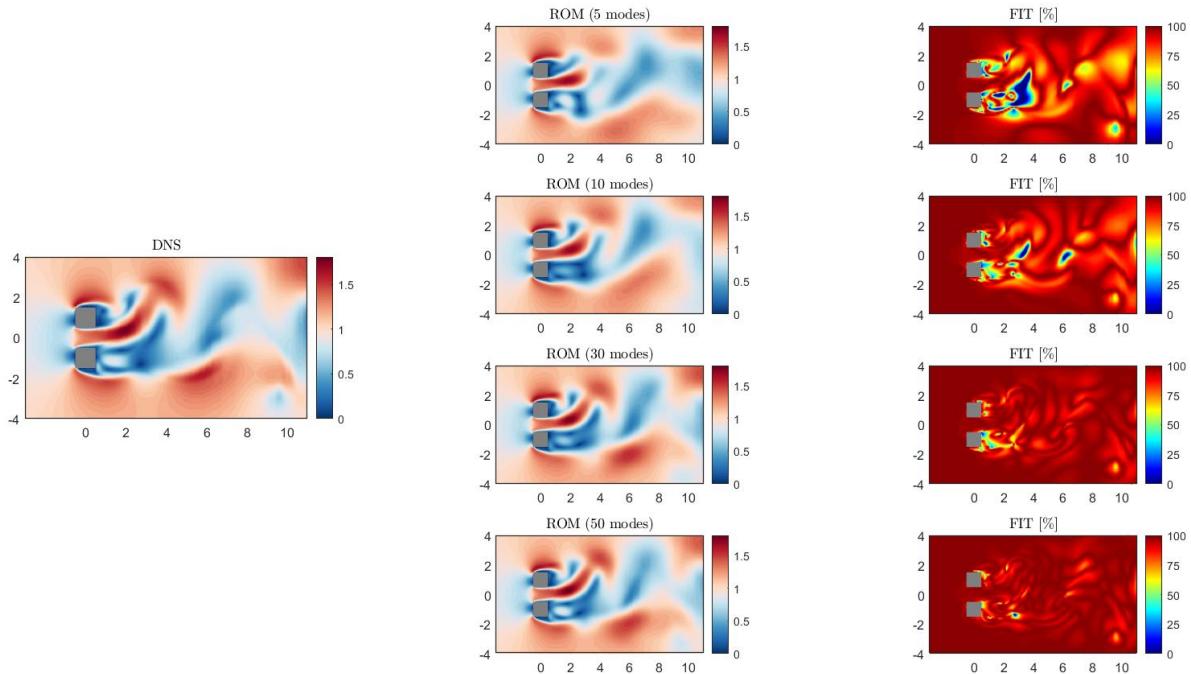


Figure 15: Reduced order model of the flow domain using 5, 10, 30 and 50 POD modes

The objective now is to use this reduced order model to design an estimator for the u' and v' fluctuations and design a controller.

4 Estimation

4.1 Galerkin projection

The application of the POD onto the DNS allows for better control robustness and responsiveness because of the lower order, more tractable system [4] [5]. We can recall from the previous chapter that 50 modes were chosen and so the temporal coefficients and spatial modes matrices can be expressed as:

$$\mathbf{a}(t) = \begin{bmatrix} a_1(t) \\ a_2(t) \\ \vdots \\ a_{50}(t) \end{bmatrix} \in \mathbb{R}^{50 \times M}, \quad \phi(\mathbf{x}) = [\phi_1(\mathbf{x}) \phi_2(\mathbf{x}) \dots \phi_{50}(\mathbf{x})] \in \mathbb{R}^{2n \times 50} \quad (13)$$

By taking advantage of the orthonormal and divergence free properties of POD, we can take the governing equation of the fluctuations (found by subtracting the RANS equation from the Navier-Stokes) onto the reduced order space given by the POD expansion. The governing equations can be separated into linear and non-linear terms as shown below

$$\nabla \cdot \mathbf{u}' = 0 \quad (14a)$$

$$\frac{\partial \mathbf{u}'}{\partial t} + \mathbf{u}' \cdot \nabla \bar{\mathbf{u}} + \bar{\mathbf{u}} \cdot \nabla \mathbf{u}' + \nabla p' - Re^{-1} \nabla^2 \mathbf{u}' = \overline{\mathbf{u}' \cdot \nabla \mathbf{u}'} + \mathbf{u}' \cdot \nabla \bar{\mathbf{u}} \quad (14b)$$

The terms on the left hand side are linear terms and the terms on the right are non-linear, where $\mathbf{u}' = [u', v']^T$. A more complete derivation of this projection is presented by Juan Guzmán-Iñigo et al. [32]. After the Galerkin projection, the evolution equation for the temporal coefficients can be found as:

$$\frac{d\mathbf{a}}{dt} = \mathcal{A}\mathbf{a}(t) + \mathcal{F}(t) + e(t) \quad (15)$$

where \mathcal{A} comes from projecting the linear terms of 14b onto $\phi_i(\mathbf{x})$, the forcing term $\mathbf{F}(t)$ from the non-linear terms and $e(t)$ from the truncation error produced from retaining m modes [5].

A state space model of \mathbf{a} is able to represent the dynamics as a combination of linear and non linear terms [5]. In discrete form, equation 15 can be presented as:

$$\mathbf{a}[k+1] = \mathcal{A}\mathbf{a}[k] + \mathcal{F}'[k] + e'[k] \quad (16)$$

The goal now is to obtain information about the underlying dynamics of \mathbf{a} . This work is required via the design of a dynamic estimator which will be discussed in the next section

4.2 Estimator design

The evolution equation of \mathbf{a} was found in he previous section. The aim is to implement an algorithm where the dynamics of equation 15 can be found. A data-driven method called system identification is used to achieve this. System identification is a powerful technique used to understand and model the behavior of complex systems. It involves the process of extracting mathematical models or representations of a system's dynamics from observed input-output data. The goal is to obtain a model that accurately captures the relationship between system inputs and outputs, allowing for predictions,

control, and analysis. The identified models are validated by comparing their predictions to the actual measured data to assess the accuracy and reliability of the identified model. For our case, the evolution equation of \mathbf{a} was found in the previous section and so we seek to implement an algorithm where the dynamics of equation 15 can be found. The system identification algorithms, namely CVA, MOESP, subspace splitting, and **n4sid**, can be unified according to Van Overschee and De Moor's theorem. This theorem proposes that all of these algorithms can be elucidated through the application of singular value decomposition to a weighted matrix, followed by a least squares estimation of the state space matrices [33] [34].

4.2.1 Subspace System Identification: **n4sid**

Research has demonstrated that employing a subspace system identification algorithm to directly extract a linear dynamical system from data and utilizing Kalman filter theory to construct a linearly optimal estimator yielded superior accuracy, robustness and computational efficiency compared to estimator matrices derived directly from the subspace identification [5]. Therefore, this project will opt to use the same methodology in designing the estimator as in [5].

There are two steps as described above, to obtain the Kalman estimation plant. The first step which is the system identification will be detailed in this section. The identification will be performed using only the output data, the temporal coefficients, with the training and validation dataset partitioned in a 50:50 split.

System identification will aim to see a linear model of the form

$$\mathbf{a}[k+1] = \mathbf{A}\tilde{\mathbf{a}}[k] + \tilde{\mathbf{f}}[k] \quad (17)$$

where $\tilde{\mathbf{f}}$ is a noise term. The model can be written more generally as

$$\mathbf{x}[k+1] = \mathbf{Ax}[k] + \mathbf{w}[k] \quad (18a)$$

$$\mathbf{a}[k] = \mathbf{Cx}[k] + \mathbf{v}[k] \quad (18b)$$

The internal state $\mathbf{x}[k]$ is introduced, where $\mathbf{x} \in \mathbb{R}^{N_x}$ and N_x is the model order chosen for the system identification. Note that the model order and the number of modes, m can be different. Equation 18a and 19 represents a linear, discrete, time-invariant system in process state-space form. The plant has matrices $\mathbf{A} \in \mathbb{R}^{N_x \times N_x}$, no input (i.e. $\mathbf{B} = 0$), $\mathbf{C} \in \mathbb{R}^{m \times N_x}$ and zero feed-through (i.e. $\mathbf{D} = 0$). The process and output noise is represented by $\mathbf{w}[k]$ and $\mathbf{v}[k]$, respectively. The noise covariances are defined as :

$$\mathbb{E} \left[\begin{pmatrix} \mathbf{w}[k_p] \\ \mathbf{w}[k_p] \end{pmatrix} (\mathbf{w}^T[k_q] \mathbf{v}^T[k_q]) \right] = \begin{pmatrix} \mathbf{Q} & \mathbf{R} \\ \mathbf{R}^T & \mathbf{N} \end{pmatrix} \delta_{pq} \geq 0 \quad (19)$$

We can notice that the true model is represented by equation 16 whilst the model to be found is shown in equation 17. In [5], it is explained that depending on if the fluctuations $\mathbf{u}'(\mathbf{x}, t)$ are infinitesimal or not, careful physical interpretation is needed. For the present case, $\mathcal{F}_i(t) \neq 0$, and therefore an eddy-viscosity approximation is introduced from the system identification which aims to approximate the true $\mathcal{F}_i(t)$ with a linear version to optimally match the approximate $\tilde{\mathbf{a}}[k]$ with the true \mathbf{a} . This proves to be advantageous since the functional form of the eddy-viscosity is generally not known. All error from the approximation are included in $\tilde{\mathbf{f}}[k]$.

The **n4sid** system identification algorithm is now used to identify the matrices (\mathbf{A}, \mathbf{C}). The result of the MATLAB **n4sid** is an IDSS structure which is represented in innovation form. The plant innovation form can be formed as below:

$$\mathbf{x}[k+1] = \mathbf{Ax}[k] + \mathbf{Ke}[k] \quad (20)$$

$$\mathbf{a}[k] = \mathbf{Cx}[k] + \mathbf{e}[k] \quad (21)$$

where the innovation vector presented as $\mathbf{e}[k]$ [5]. The innovation vector is related to the process noise via:

$$\mathbf{w}[k] = \mathbf{Ke}[k] \quad (22)$$

where \mathbf{K} is a gain matrix. A complete derivation of the steps that start from the process state-space form to the innovation form can be found in [33]. The covariance of the process noise is defined as $\mathbf{Q} = \text{cov}(\mathbf{w}) = \mathbb{E}(\mathbf{ww}^T)$ and as described in [5], and computed by

$$\mathbf{Q} = \text{cov}(\mathbf{Ke}) = \mathbf{K}\text{cov}(\mathbf{e})\mathbf{K}^T = \mathbf{K}\mathbb{E}(\mathbf{ee}^T)\mathbf{K}^T \quad (23)$$

4.2.2 Kalman filter

The next step is to design the optimal kalman filter estimator by adding the sensor measurements. An assumption is made that the sensors can record only a single velocity component. Mentioned earlier in the DNS chapter, the presentation grid is a type of probe, where probe and sensors are used interchangeably for this context. The presentation grid contained 11,259 probes and so therefore there is a maximum of 22,518 sensors given that each point measures the component of velocities separately. The sensors will be stored in a vector $\mathbf{s}[k]$, and it can be written in terms of the POD temporal coefficients given by [5]:

$$\mathbf{s}[k] = \mathbf{Sa}[k] + \mathbf{g}[k] \quad (24)$$

where matrix \mathbf{S} is formed by taking the rows of $\phi(\mathbf{x})$ (refer back to equation 8) that relates to the coordinate \mathbf{x}_j and velocity component recorded at each respective location and $\mathbf{g}[k]$ is the noise term that comes from the POD truncation. The matrix form of $\mathbf{S} = \mathbb{S}_\phi \phi(\mathbf{x})$, where \mathbb{S}_ϕ is the row selection matrix with rows equal to the number of sensors chosen, p , and $2n$ columns, where it is reminded that $n = 11,259$ for this case. A more detailed explanation of the \mathbb{S}_ϕ matrix is described in [5]. We can now find the noise covariance \mathbf{R} by

$$\mathbf{R} = \mathbb{E}(\mathbf{gg}^T) \quad (25)$$

where \mathbf{g} can be found by rearranging equation 24. A question to consider is the the number of sensors chosen and sensor location. It is ideal to have the optimal sensor locations for the best performance of the Kalman filter. This will be discussed in the later sections.

Now, all the necessary components are prepared to formulate the Kalman filter estimator as given below:

$$\hat{\mathbf{x}}[k+1] = \mathbf{A}\hat{\mathbf{x}}[k] + \mathcal{L}(\mathbf{s}[k] - \hat{\mathbf{s}}[k]) \quad (26a)$$

$$\hat{\mathbf{a}}[k] = \mathbf{C}\hat{\mathbf{x}}[k] \quad (26b)$$

$$\hat{\mathbf{s}}[k] = \mathbf{S}\hat{\mathbf{a}}[k] = \mathbf{SC}\hat{\mathbf{x}}[k] \quad (26c)$$

where any terms with a (\cdot) is the estimated version of the same term and \mathcal{L} is the Kalman filter gain. \mathcal{L} can be obtained by using the MATLAB in-built function `kalman` that takes in the plant and the covariances, \mathbf{Q} and \mathbf{R} , that works by solving the Riccati equation [35].

4.2.3 QR pivoting algorithm

The quality of reconstruction is influenced by factors such as the model order, N_x , and the number and positioning of the sensors, p . [5]. It is said that finding optimal sensors is an NP-hard problem [4], in other words there are $\binom{N}{P} = \frac{N!}{(N-P)!}$ combinations of P sensors at N locations where P and N is translated to p and $2n$, respectively. The solution to this problem can become intractable as the number of sensors are increased and therefore a sub-optimal solution is required. A QR-pivoting algorithm is an optimal sensor method detailed that will be used. We can consider the fully sensed (i.e. $\mathbb{S}_\phi = \mathbb{I}_{2n \times 2n}$, discrete time system, \mathbf{G} :

$$\hat{\mathbf{x}}[k+1] = \mathbf{A}\hat{\mathbf{x}}[k] + \mathbf{K}\mathbf{e}[k] \quad (27a)$$

$$\mathbf{s}[k] = \phi\mathbf{a}[k] = \phi\mathbf{C}\hat{\mathbf{x}}[k] \quad (27b)$$

The full detail of the methodology can be found in [5][36] and the definition of the \mathbb{H}_2 is introduced [37]. We can again consider a partially sensed system like with the Kalman filter such that the output is again $\mathbf{s}[k] = \mathbf{S}\hat{\mathbf{a}}[k] = \mathbb{S}_\phi\phi\mathbf{C}\hat{\mathbf{x}}[k]$. The optimal sensor selection involves a Cholesky decomposition of the output controllability Gramian (i.e. $\mathbf{W}_{oc} = \mathbf{F}\mathbf{F}^T$) for $N_x \gg m$ and letting the $p = m$ [5]. \mathbf{S} can be solved by applying the QR factorisation to $(\phi\mathbf{F})^T$ by:

$$(\phi\mathbf{F})^T \mathbf{P} = \mathbf{Q}\mathbf{R} \quad (28)$$

where P is the permutation matrix, Q is a unitary matrix and R is an upper-triangular matrix. The first p entries contains the indices of the best ranked sensors from best to worst. The output controllability Gramian \mathbf{W}_{oc} was shown to be defined as $\mathbf{C}\mathbf{W}_c\mathbf{C}^T$ where \mathbf{W}_c is the controllability Gramian. The in-built MATLAB function `dlyap` was used to find the controllability Gramian which is calculated as the solution of the Lyapunov equation as $\mathbf{A}\mathbf{W}_c\mathbf{A}^T - \mathbf{W}_c + \mathbf{K}\mathbf{K}^T = 0$. Once the sensors have been selected, the number of sensors can be selected to be $p_2 < p = m$.

4.3 Design overview

The whole process of the design of the estimator can be summarised with the following steps

1. Simulate the flow model on a CFD software and obtain the DNS data
2. Form the snapshot matrix \mathbf{Y} and derive the modes ϕ along with the temporal coefficients $\mathbf{a}(t)$
3. Use the output data (the temporal coefficients $\mathbf{a}(t)$) and feed it into the `n4sid` system identification algorithm to retrieve the matrices (\mathbf{A}, \mathbf{C}) and use the matrix \mathbf{K} to obtain the \mathbf{Q} and \mathbf{R} covariances
4. Employ the QR pivoting algorithm to find the optimal sensor placement which involves the calculation of the output controllability Gramian \mathbf{W}_{oc} , and use it to assemble the measurement vector $\mathbf{s}[k]$
5. Obtain the Kalman filter gain \mathcal{L} by using the matrices $\mathbf{A}, \mathbf{C}, \mathbf{S}, \mathbf{Q}, \mathbf{R}$ to solve the Riccati equation

The schematic of the estimator can be visualised on figure 16.

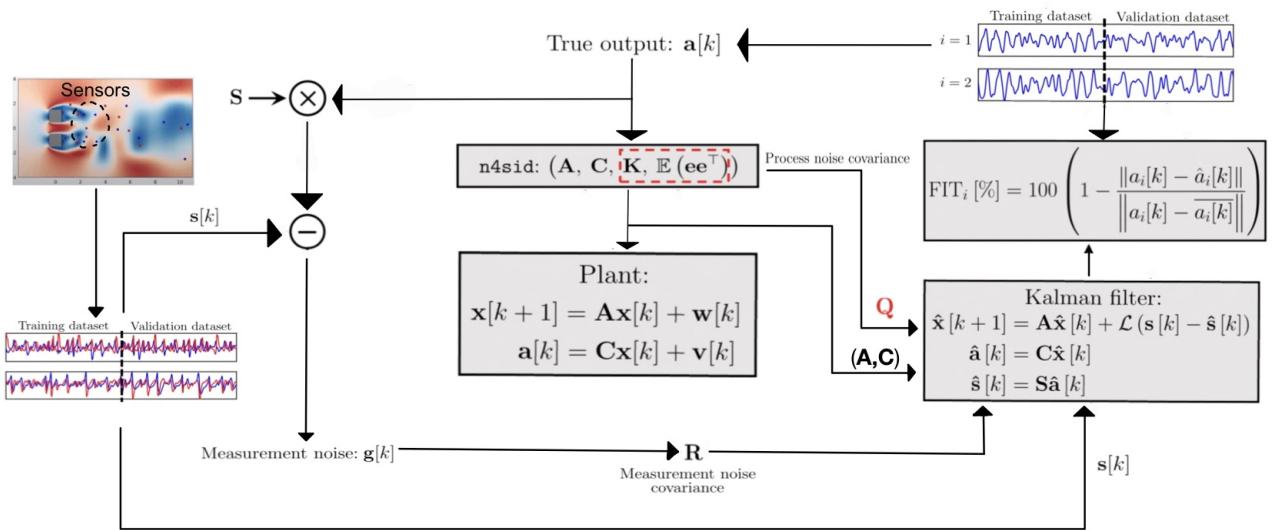


Figure 16: Diagram illustrating the utilization of system identification and Kalman filter in the estimator

4.4 Estimator performance

The final estimator was designed using model order, $N_x = 50$ and the number of POD modes, $m = 50$. The numbers of sensors, p chosen was 20. The temporal coefficient signal can be found on figure 17, where the blue lines are the true coefficients and the red lines are the estimated temporal coefficients, \hat{a} . It can be seen that for the first ten modes, the estimation follows the true signal to a good level of accuracy as the time average FIT[%] does not fall below 68%. As the higher modes are estimated (modes 11-50 not shown), the FIT[%] shows a gradual decrease, however this should not be a major concern as it should be recalled that the higher modes contains less energy of the system.

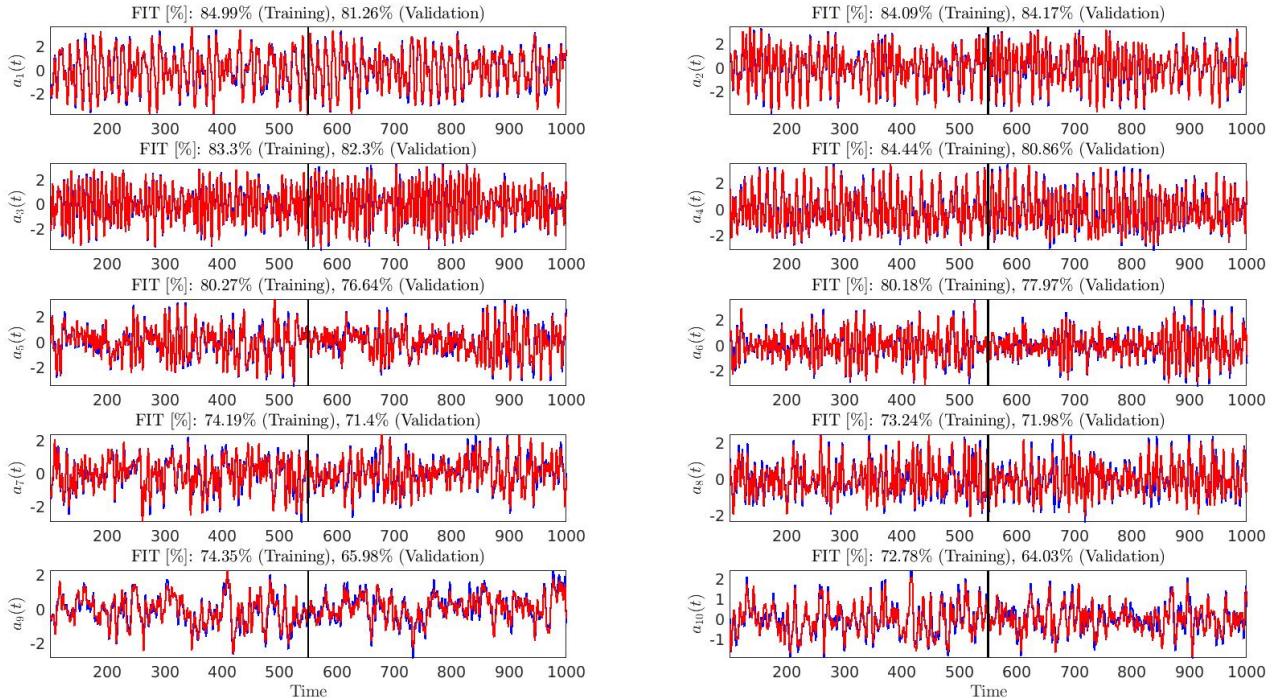


Figure 17: FIT[%] for the first 10 POD temporal coefficients using 20 sensors

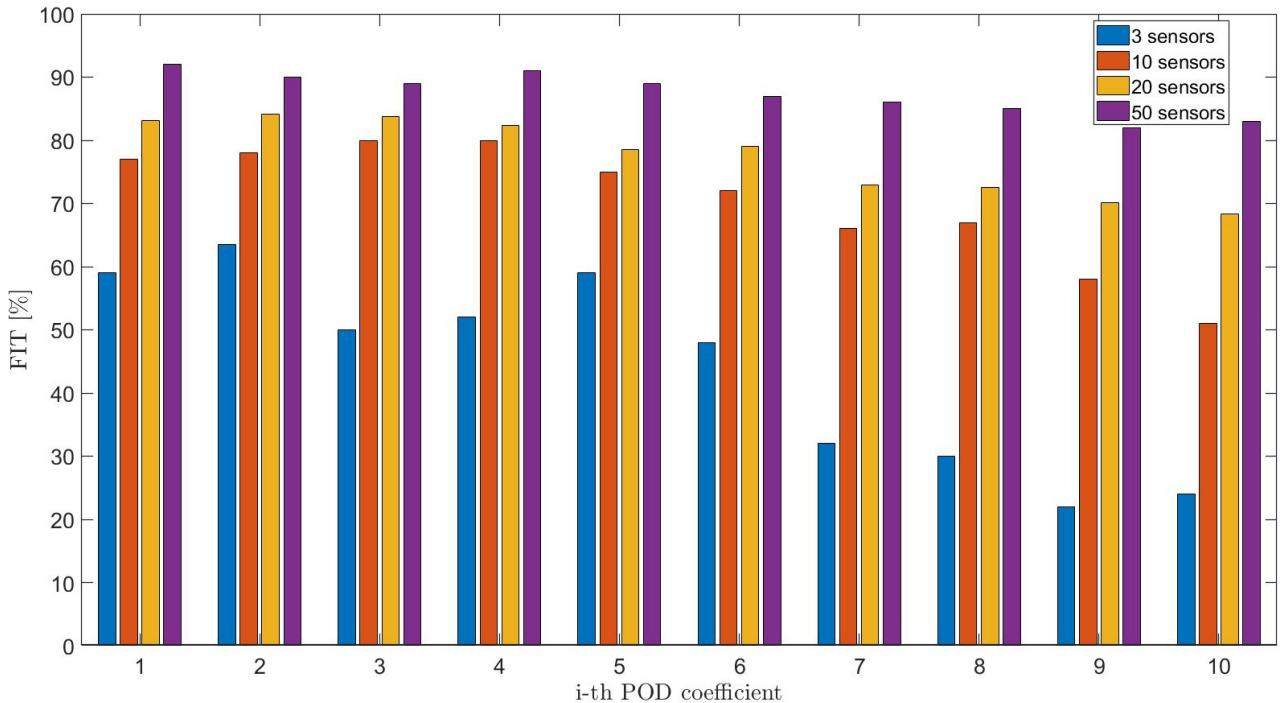


Figure 18: FIT[%] of POD coefficients for different number of sensors

A comparison of the estimation performance against the number of sensors was conducted. It is clear that as the number of sensors are increased, the estimation of the flow is improved. It was found that the estimation of the wake is improved when increasing the numbers of sensors from 10 to 20. This is due to more sensors being located at the rear region of the flow domain. Figure 18 illustrates the performance increase with the number of sensors. It should be noted however, that using more sensors may not be suitable experimentally like in PIV. Additional sensors can introduce more noise and measurement errors into the system. By selecting fewer sensors, it is possible to minimize the impact of noise and focus on the most relevant measurements. This can simplify the system model, reduce computational requirements, and enhance the robustness and reliability of the overall system.

A short study was conducted to compare the accuracy of the 'optimal' sensors and the sensors found at POD peaks. (sensor locations at POD peaks is not discussed and the reader should look at [5] for more information). The number of sensors selected for comparison was 20. Figure 19 implies that the QR pivoting algorithm is superior in estimating the flow characteristic for all modes, and as the modes increases the difference in performance widens. This result reinforces the use of the greedy QR pivoting algorithm for sensor selection and was explained by Savarino that the reason was because the sensors were more spread out for QR than POD peaks [5].

The estimation of different regions of the flow is analysed. The estimator was able to detect the underlying structure of the flow proven by the close relation at the spectra peaks. The estimation for $x = 1$ and $x = 4$, was exceptionally executed with small regions that do not overlap. At $x = 10$, the estimator performed slightly worse for both the u' and v' spectra. This is likely due to most sensors being located just rear of the prisms or in the middle of the flow domain and few located in the far wake. A video version of the estimation of the flow using a Kalman filter with specifications of: model order $N_x = 50$, $m = 50$ and $p = 20$ though a link on figure 22, where the red squares are the sensors for the u velocity and blue circles are the sensors for the v velocity. Interestingly, the number of sensors for each component are equal for when 20 sensors are selected. This is a dramatic change compared to when 10 sensors were selected and 9/10 sensors recorded the u velocity.

The estimator performance can also be supported through a FIT [%] plot of the whole flow domain. It can be seen on figure 21, that the estimator has a good level of accuracy for the majority of the domain with only a few small blue patches where the estimator was not able to reconstruct the flow to a good level.

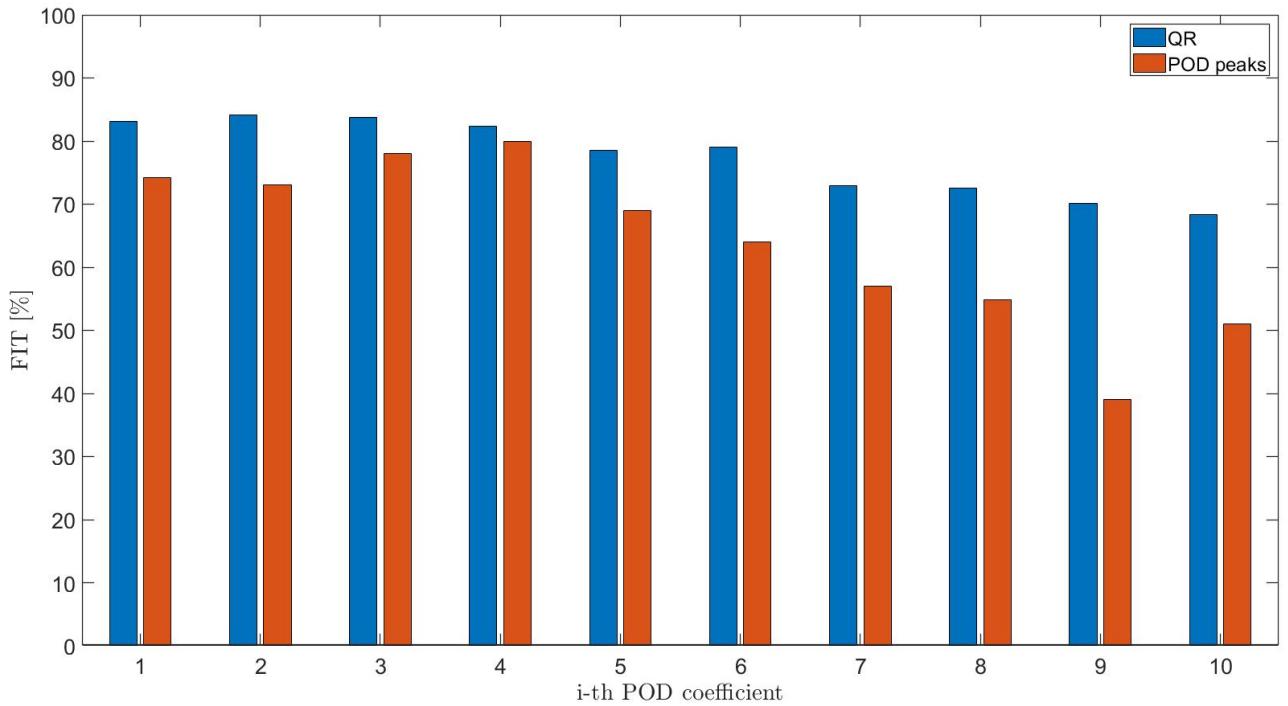


Figure 19: QR pivoting algorithm vs POD peaks estimation performance

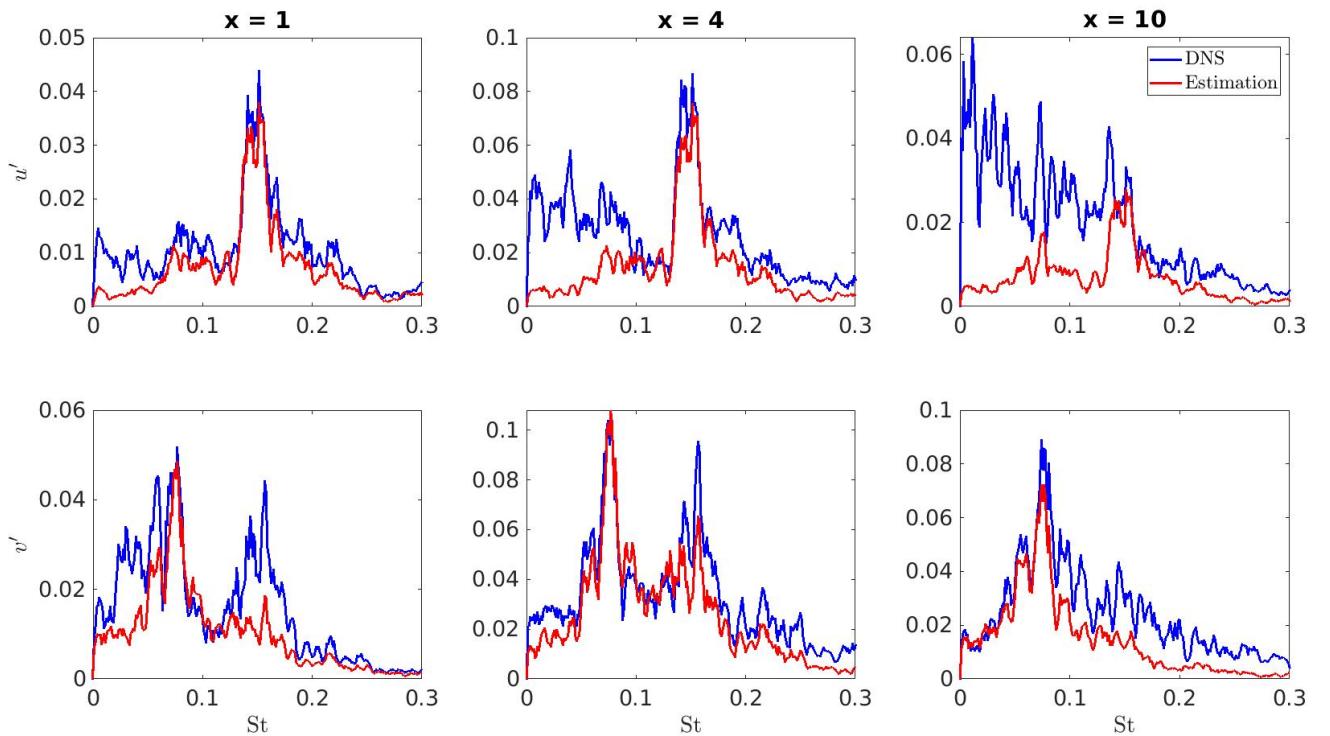


Figure 20: Reconstructed vs DNS velocity spectra at different probe points

The completion of the estimator design now leads to the controller design which will be discussed in the next chapter.

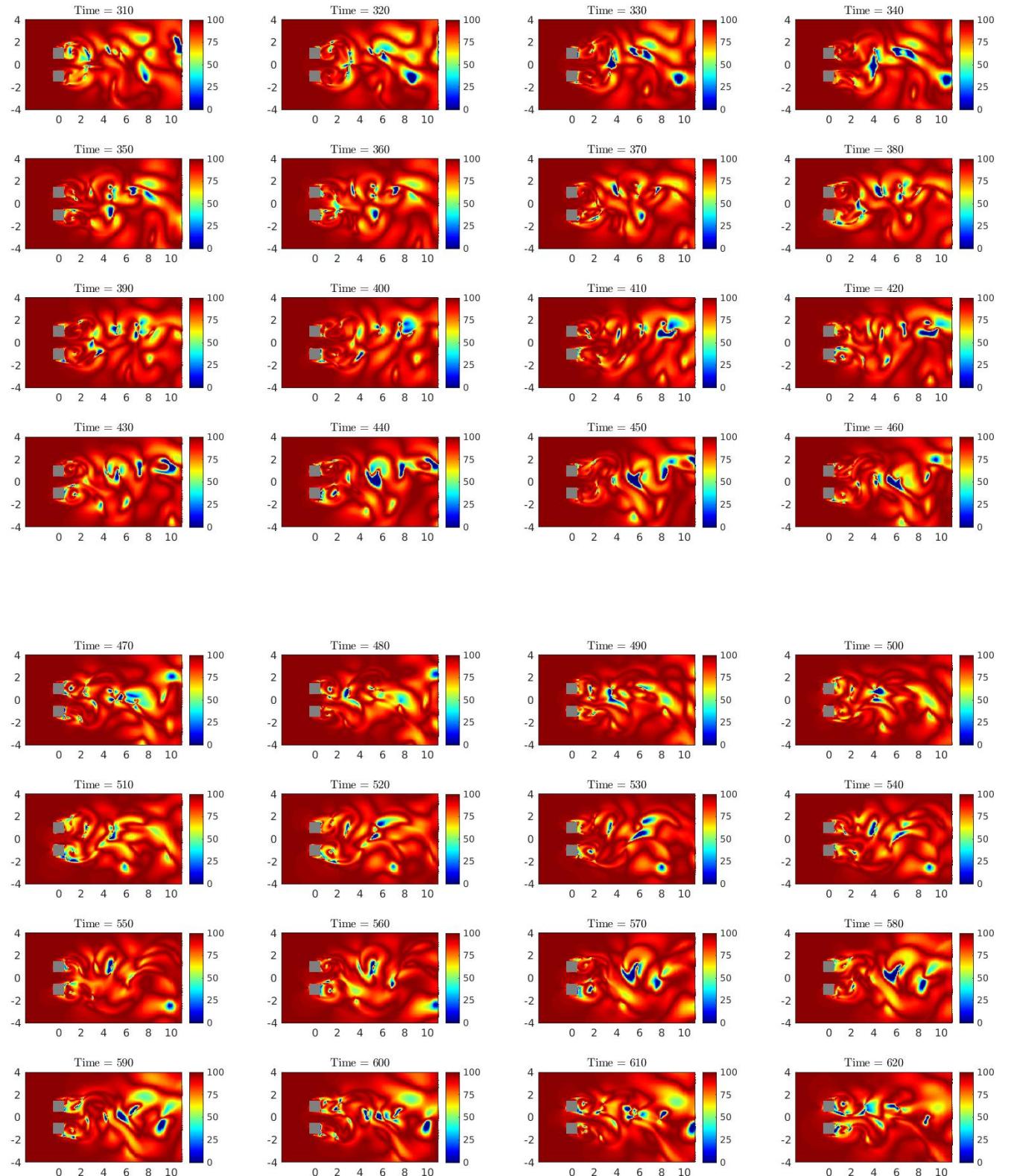


Figure 21: FIT [%] of flow at different time instances

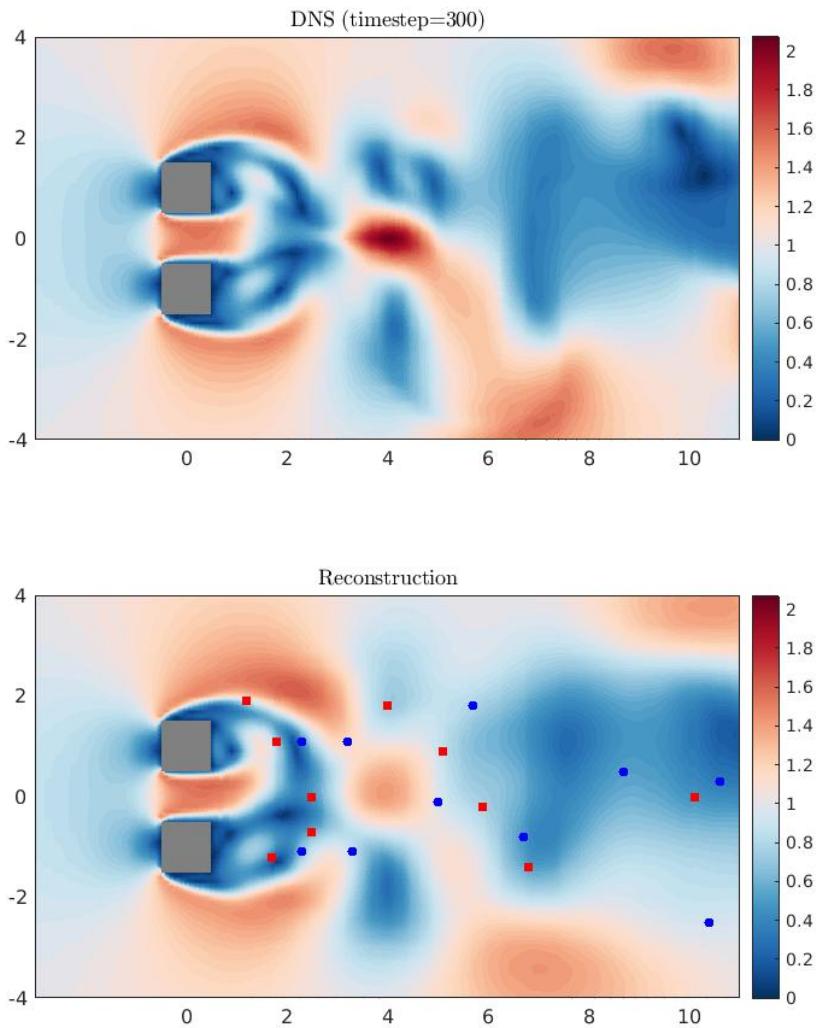


Figure 22: Snapshot of the estimation at timestep = 300 for a model order $N_x = 50$, number of modes $m = 50$, and $p = 20$. A video version of this flow simulation can be found online at: <https://www.youtube.com/watch?v=YFide5iFmFM>

5 Control

The LQG (Linear Quadratic Gaussian) controller is widely used in control systems due to its effectiveness in achieving optimal control performance. The controller combines two key components: the LQR (Linear Quadratic Regulator) controller and the Kalman filter [38]. The LQR controller provides an optimal control strategy by minimizing a quadratic cost function that captures desired performance criteria, such as tracking error or control effort. By formulating the control problem in a quadratic optimization framework, the LQR controller generates control inputs that balance between achieving desired system performance and minimizing control effort. By integrating the LQR controller and the Kalman filter, the LQG controller optimally combines control action and state estimation. The controller utilizes the estimated state information to generate control inputs that actively compensate for disturbances and uncertainties, leading to improved control performance and robustness. With the reduced-order model and the estimator in place, the next step is to design the controller. The estimator is responsible for accurately estimating the system's state based on available measurements, while the controller generates the optimal control input using the estimated state.

5.1 Controller design

5.1.1 System excitation

A closed-loop control method is considered to suppress the vortex shedding. Consider the system introduced in chapter 3, where it can be altered to include control as seen below:

$$\mathbf{x}[k + 1] = \mathbf{A}\mathbf{x}[k] + \mathbf{B}\mathbf{u}[k] + \mathbf{w}[k] \quad (29a)$$

$$\mathbf{a}[k] = \mathbf{C}\mathbf{x}[k] + \mathbf{v}[k] \quad (29b)$$

where \mathbf{B} is the input matrix, and $\mathbf{u}[k]$ is the known input signal. Following Hervé et al [39], the system was excited with a frequency-rich signal to identify the input matrix. It should be noted that the excitation signal should contain frequencies similar to frequencies that the system exhibits. By using an excitation signal with a PSD that covers the range of frequencies that exists in the flow, the system's behaviour will be stimulated. The excitation signal should cover both low and high frequencies that may affect the system's response. It should have sufficient energy across different frequency bands to provoke a representative response from the system. If the system exhibits specific frequency-dependent behaviors, such as resonances or dominant modes, it is crucial to include those frequencies in the excitation signal. Following Guzmán-Iñigo [10], the excitation signal was selected to be a random binary signal (RBS), which excites the characteristics frequencies. This can be created on MATLAB using `idinput`, to create input signals and selecting the 'rbs' type for RBS. The signal will be fed through at the actuator locations where the actuation should be zero mean average so the signal does not introduce a persistent offset or bias in the system's response which means the actuator will perform blowing and suction of the flow. This allows the controller to focus on regulating the system around its desired setpoint without being affected by any inherent bias in the actuation signal. Figure 23 illustrates the PSD of the excitation signals where it shows similarities with the PSD of the flow, noticeably peaks exists at St = 0.05 and 0.15.

The actuator location was selected to be at the edge of the prisms with a single actuator on each horizontal surface of the prisms as the offset of vortex shedding is at the vertices of the prisms. The actuator width should be wide enough to affect the flow where a width of 25% of prism length is chosen. The locations of the actuators can be shown on figure 24, where the horizontal blue lines are the actuators and the arrows on the figures shows the direction of the flow when the excitation signal is positive. If the arrows are pointing towards the prisms, this means the signal is negative and there will be suction of flow towards the prism. On STAR-CCM+, the actuator can be created by creating an imprint in the CAD model and setting this surface as a 'Velocity Inlet' boundary condition. The

excitation signal is fed at the location of the actuator by importing the signal in CSV format and using Field functions at the velocity inlet.

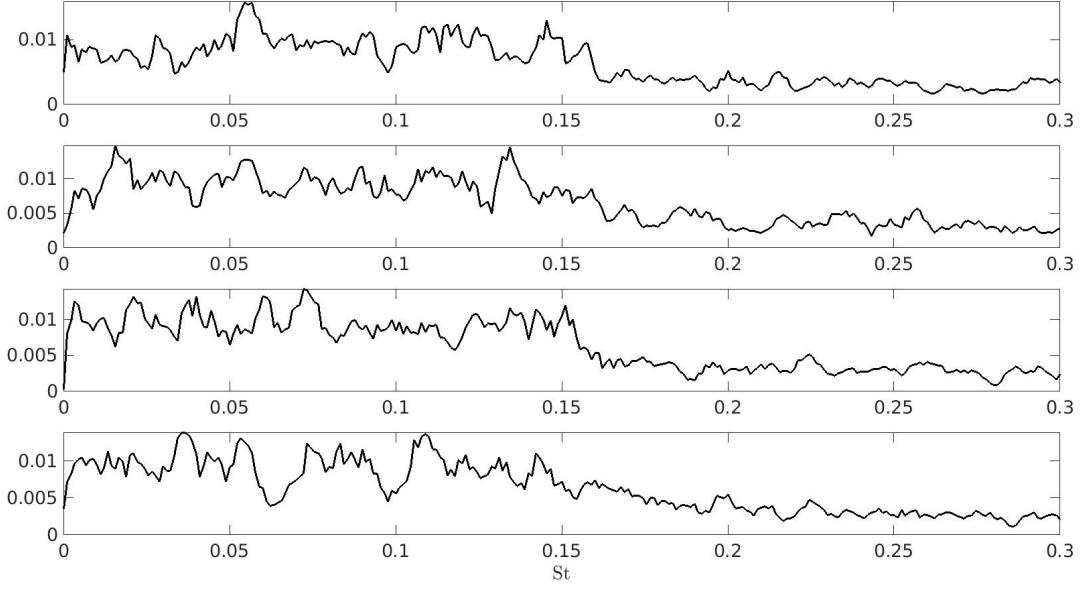


Figure 23: PSD of excitation signals for the four channels

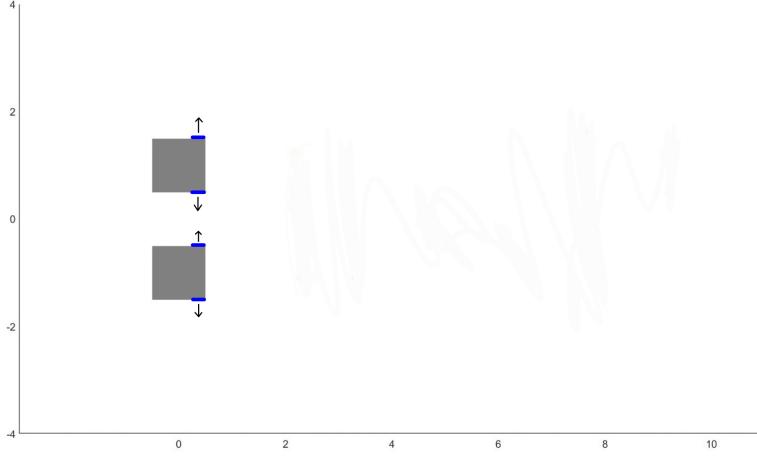


Figure 24: Location of actuators

Now that the input signal is obtained, it can be used to obtain the excited characteristic of the system and the synthesis of one half of the LQG can be made. The procedure will follow a very similar one to when the estimator was designed in 4.3. The modified procedure is as follows:

1. Simulate the flow model with the excited signal on a CFD software and obtain the DNS data
2. Form the snapshot matrix \mathbf{Y} and derive the modes ϕ along with the temporal coefficients $\mathbf{a}(t)$ for the new model
3. Use input signal and the output data (the temporal coefficients $\mathbf{a}(t)$) and feed it into the `n4sid` system identification algorithm to retrieve the matrices $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ and use the matrix \mathbf{K} to obtain the \mathbf{Q} and \mathbf{R} covariances
4. Use the sensors found in chapter 4

5. Obtain the Kalman filter gain \mathcal{L} by using the matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{S}, \mathbf{Q}, \mathbf{R}$ to solve the Riccati equation

The modified kalman filter is synthesised and now the development of the LQR is needed before the closed-loop system can be implemented.

5.1.2 Linear Quadratic Gaussian

The LQG is a controller that consists of the Linear Quadratic Regulator (LQR) and the estimator (Kalman filter). It has previously been implemented for control of an open cavity flow [22] and noise amplifier flows [10]. In the latter, the controller was implemented with the control objective of minimising the perturbation energy of the system and was proved to be successful. The aim is to adopt this control objective and extend it to see if the perturbation energy (vortex shedding) can be damped to decrease the drag for highly non-linear chaotic flow. Here the synthesis of the controller will be shown. The LQR is commonly known in the control systems community as a linear state space optimal controller with a control law, $u = -\mathbf{K}\mathbf{x}$, where \mathbf{K} is the optimal gain matrix and \mathbf{x} is the state vector, which minimises the cost functional in discrete form:

$$J = \sum_{k=0}^{\infty} (x^T Q x + u^T R u + 2x^T N u) \quad (30)$$

where Q is the state cost weighted matrix, R is the input cost weighted matrix and N is the cross term matrix. For the system considered, there is no feed through (i.e. $\mathbf{D} = 0$), and therefore the cross term matrix is automatically zero. Additionally, equation (30) is shown, as commonly the state is what is regulated. However, the control objective considered the minimisation of the POD modes which are related to the temporal coefficients \mathbf{a} . Therefore, the cost functional equation will need to be modified to represent this change. Lets first change the equation so we are regulating the output and remove the cross term, so we now get:

$$J = \sum_{k=0}^{\infty} (a^T Q a + u^T R u) \quad (31)$$

we know that from our state space equation that $\mathbf{a} = \mathbf{Cx}$, and therefore we can reach the modified equation

$$J = \sum_{k=0}^{\infty} ((\mathbf{Cx})^T Q (\mathbf{Cx}) + u^T R u) = \sum_{k=0}^{\infty} \left(x^T (\underbrace{\mathbf{C}^T Q \mathbf{C}}_{Q^*}) x + u^T R u \right) \quad (32)$$

where Q^* is the new state weighting matrix and R is the same input cost weighted matrix as before. In [10], Q was set as the identity matrix, \mathbf{I} , to suppress the energy of the velocity window. So therefore, the state weighting matrix becomes $Q^* = \mathbf{C}^T \mathbf{C}$. The input cost matrix remains unchanged and is a parameter that is user defined. Increasing the values in the R matrix puts a higher penalty on the control effort, meaning that the controller will prioritize minimizing the control inputs. This can result in smoother control actions but may lead to larger deviations in the state tracking performance [40]. The control gain \mathbf{K} is obtained by solving the Riccati equation involving $\mathbf{A}, \mathbf{B}, Q^*$ and R . It should be noted that the control law $u = -\mathbf{K}\mathbf{x}$ here involves the true state \mathbf{x} , however this information is not available and so the estimated state from the Kalman filter is used instead to formulate the control input, $u = -\mathbf{K}\hat{\mathbf{x}}$. We can now combine the LQR controller with the Kalman filter

$$\hat{\mathbf{x}}[k+1] = \mathbf{A}\hat{\mathbf{x}}[k] + \mathbf{B}\mathbf{u}[k] + \mathcal{L}(\mathbf{s}[k] - \hat{\mathbf{s}}[k]) \quad (33a)$$

$$\hat{\mathbf{a}}[k] = \mathbf{C}\hat{\mathbf{x}}[k] \quad (33b)$$

$$\hat{\mathbf{s}}[k] = \mathbf{S}\hat{\mathbf{a}}[k] = \mathbf{S}\mathbf{C}\hat{\mathbf{x}}[k] \quad (33c)$$

$$\mathbf{u}[k] = \mathbf{K}\hat{\mathbf{x}}[k] \quad (33d)$$

where \mathbf{A} , \mathbf{B} , \mathbf{C} are the system matrices from `n4sid`, \mathcal{L} is the Kalman filter gain, \mathbf{S} is the matrix containing the rows of $\phi(\mathbf{x})$. The true sensor measurement \mathbf{s} should be extracted in real time from the simulation.

The implementation of the closed loop controller on STAR-CCM+ can be achieved through the use of the Java API/ macro. All matrices were imported into the Java code and any linear algebra needed within the code was done with the Apache Commons Mathematics library.

The whole process for the controller can be summarised by the following:

1. Generate a frequency rich signal that contains a range of frequencies including the characteristic frequencies of the system
2. Obtain the DNS data for the excited system
3. Find the eigenmodes ϕ and temporal coefficients $\mathbf{a}(t)$
4. Feed the input signal and the output data (the temporal coefficients $\mathbf{a}(t)$) into the `n4sid` system identification algorithm to retrieve the matrices (\mathbf{A} , \mathbf{B} , \mathbf{C}) and use the matrix \mathbf{K} to obtain the \mathbf{Q} and \mathbf{R} covariances
5. Obtain the Kalman filter gain \mathcal{L} by using the matrices \mathbf{A} , \mathbf{B} , \mathbf{C} , \mathbf{S} , \mathbf{Q} , \mathbf{R} to solve the Riccati equation
6. Use matrices \mathbf{A} , \mathbf{B} , \mathbf{Q}^* , \mathbf{R} to solve the Riccati equation to retrieve the control gain \mathbf{K} (control gain \mathbf{K} should not be confused with matrix \mathbf{K} found from `n4sid`)
7. Combine the estimator and controller together to form the LQG and form the closed loop system on the CFD software

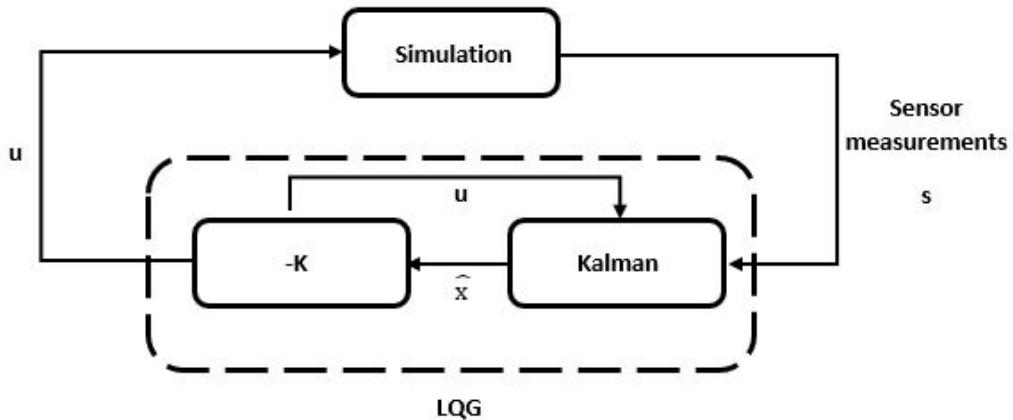


Figure 25: Schematic of LQG

5.2 Controller performance

The controller performed reasonably well in dampening the fluctuation velocities when it was well tuned. Figure 26 illustrates the effectiveness of the controller for different configurations. For the u' velocity with $R = 60$, the peak decreased by around 5%. However, beyond $x = 2.9$, the u' RMS velocity is slightly greater for the control case. The amplitude of the v' fluctuation velocity decreased by approximately 25% at $x = 4$. The v' fluctuation velocity also shifts closer to the prisms for the control case. It can be seen that after $x = 2.5$, the v' RMS velocity is lower when there is control. However, when $R = 20$, the control made the flow worse than without control. The peak for the RMS velocities is greater for both coordinate directions and the velocity is generally greater after the peak. This can be explained as the actuators' velocity would fall in regions around 1.2 m/s. This actuation is clearly too strong and provides more energy into the system that taking energy out. The actuation velocity for $R = 60$ was found to be no greater than 0.5 m/s. It should be noted that \mathbf{R} is not a scalar value but a matrix, where $\mathbf{R} = k\mathbf{I}$, and k is a scalar value and \mathbf{I} is the identity matrix, and therefore the actuators have no cross-coupling with one another. The signal of the fluctuation velocity was analysed at the peak for the non-control case, and it can be concluded that the overall fluctuations were attenuated. The signal of v' at $x = 4$, can be found on figure 27. Additionally, the PSD of the velocities, shown in figure 28, at different locations can be assessed. It was mentioned in the earlier chapters that a $St = 0.15$, is closely related to the vortex shedding of the prisms. The spectra for u' and v' clearly indicates that just rear of the prisms, the control and non-control case are very close to one another which means the controller was unable to mitigate the vortex shedding phenomenon itself however, going downstream, the v' characteristic frequencies when control is applied is shown to be attenuated insinuating that the vortex shedding affects downstream is not as effective.

The estimator performance when used within the controller was visualised through figure 29. In comparison to figure 21, there are more blue regions which shows that the estimator's performance was slightly worsened. Nonetheless, there was still large areas which showed good accuracy of the reconstruction of the flow.

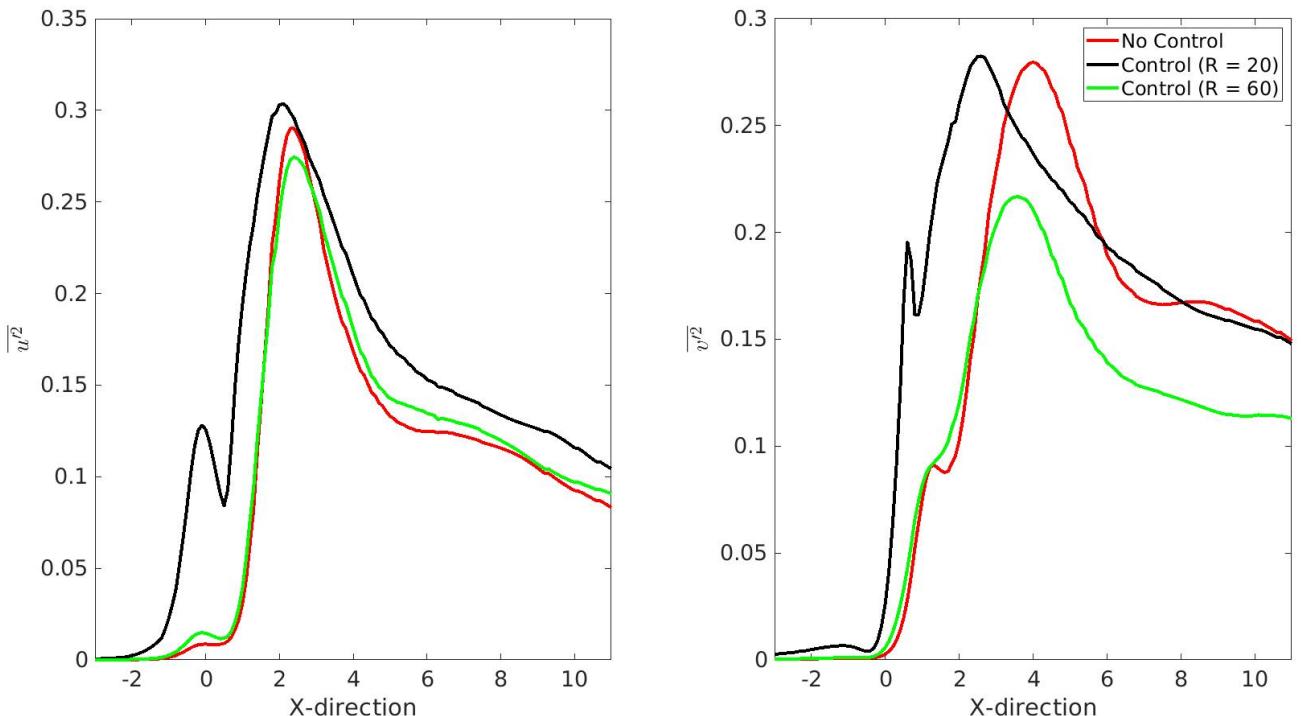


Figure 26: RMS velocity of the midline of the flow domain

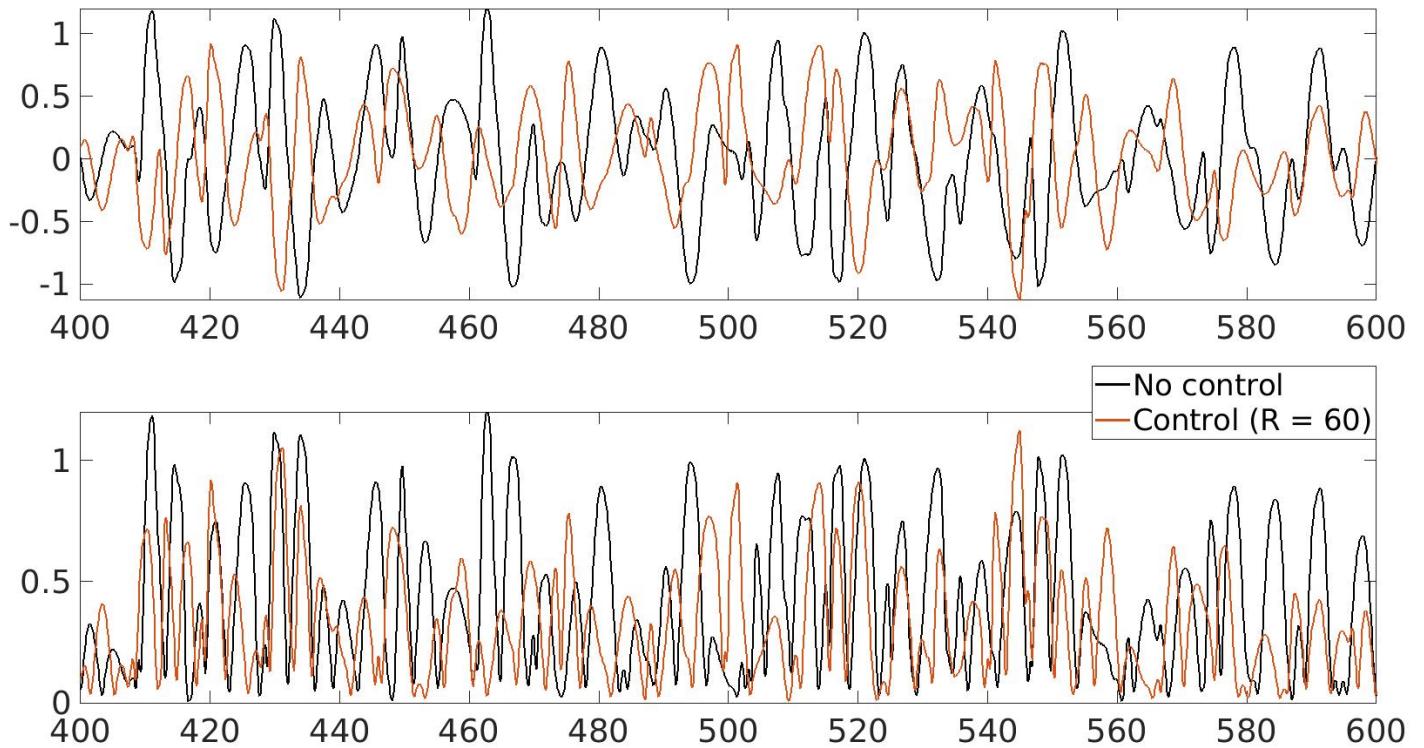


Figure 27: v' velocity (top) and v' velocity magnitude signals at $x = 4$

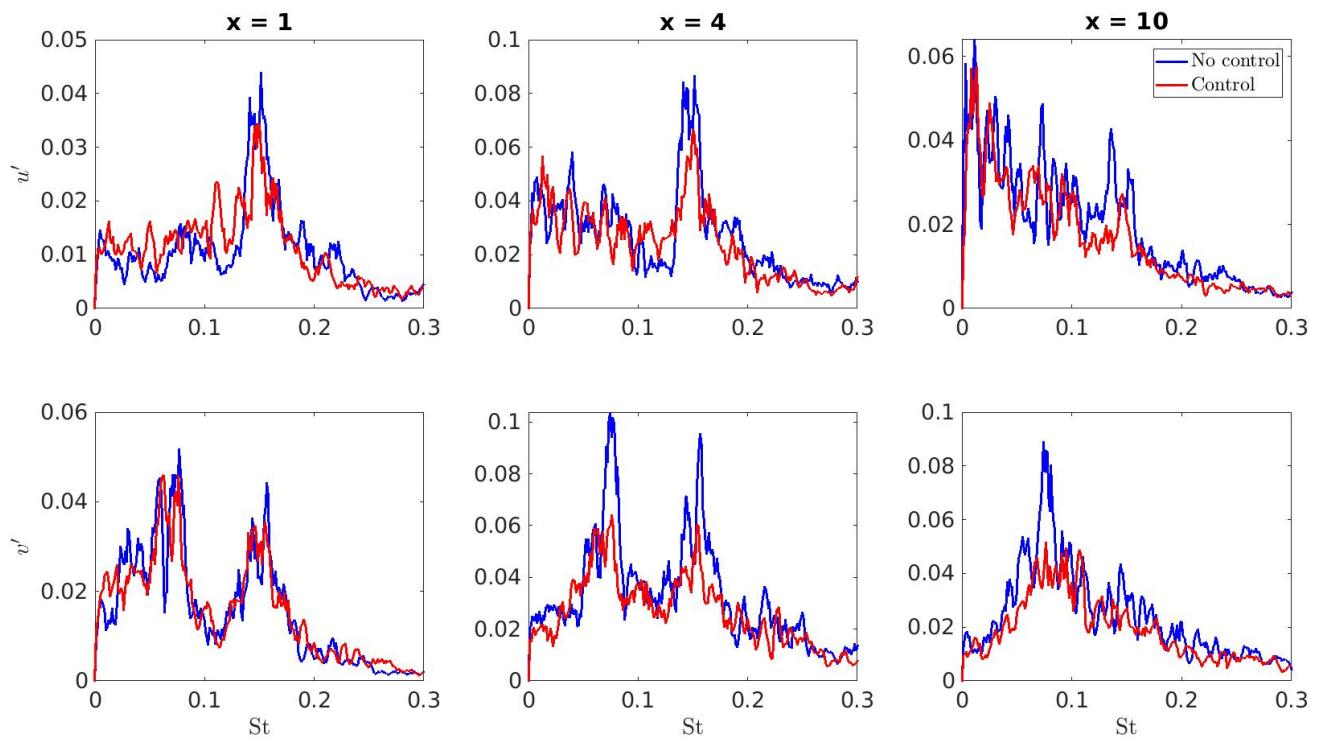


Figure 28: PSD for DNS and control at $x = 1, 4$ and 10

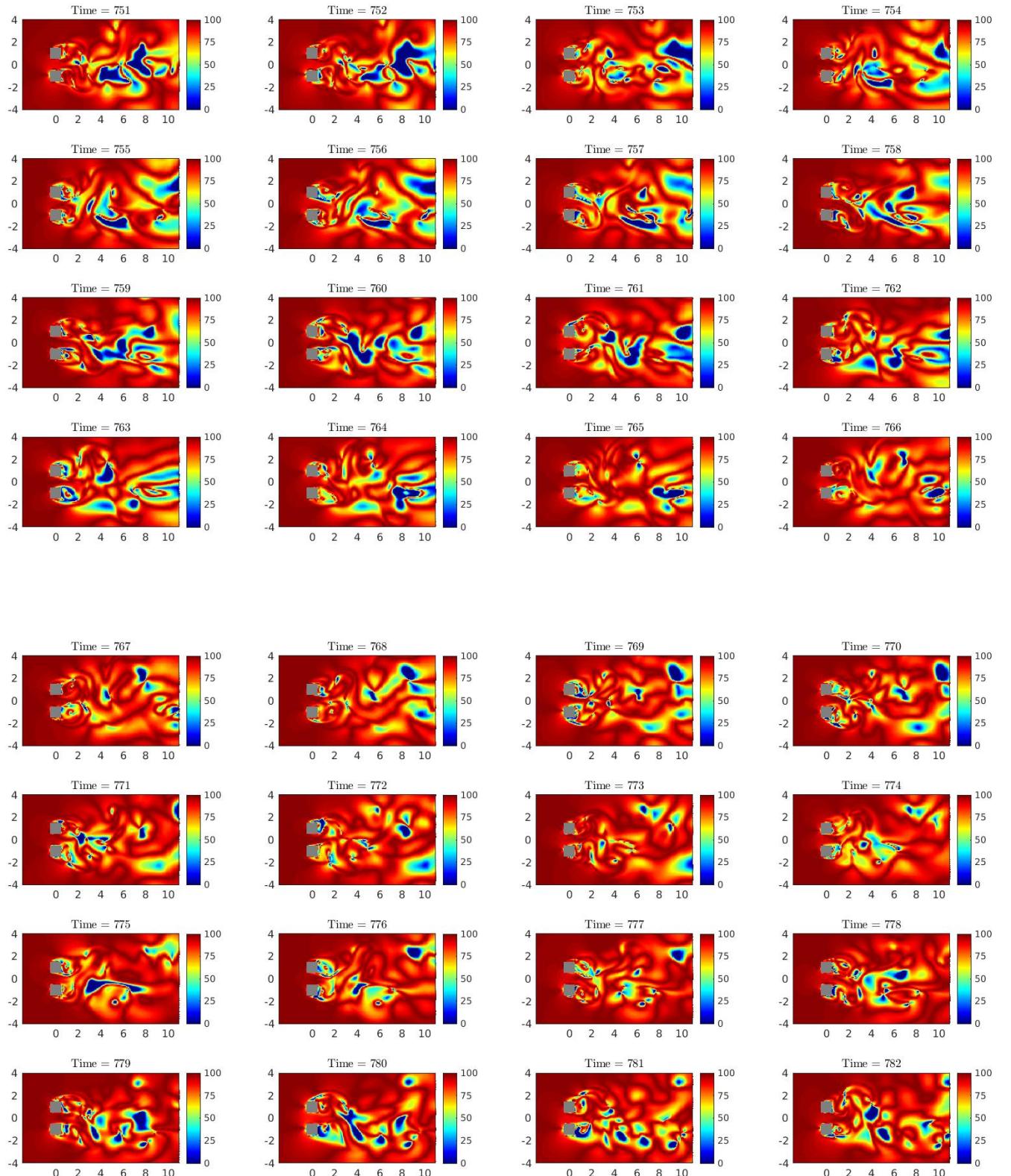


Figure 29: FIT [%] of flow at different time instances with the controller

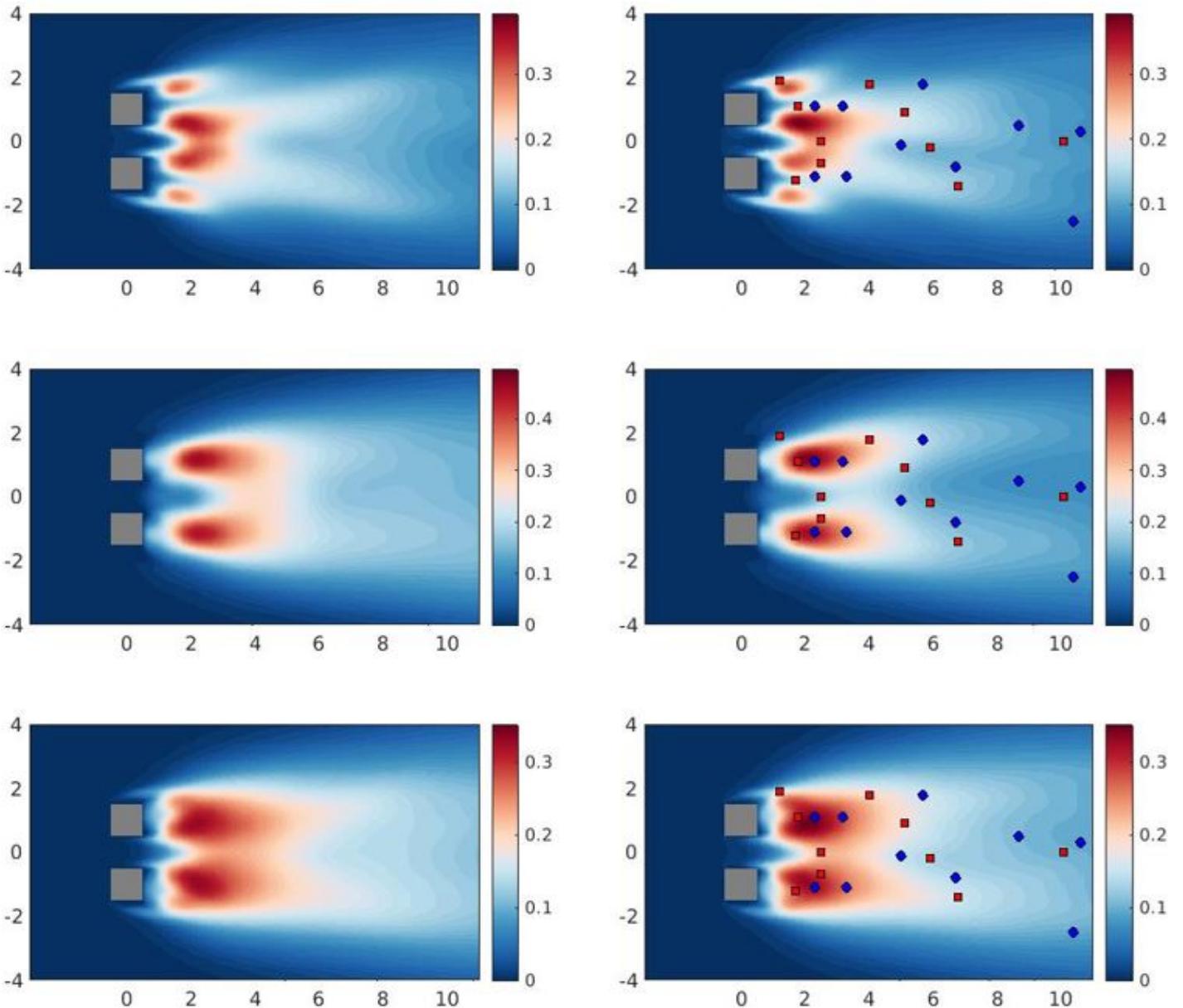


Figure 30: Contour plots of velocity statistics: (top row) $\overline{u'^2}$, (middle row) $\overline{v'^2}$, (bottom row) fluctuating kinetic energy for (left column) DNS and (right column) control, red and blue markers are the u and v velocity sensors, respectively

The contour plots for the velocity statistic is displayed on figure 30. It can be noticed that the symmetry of the flow about the centreline $y = 0$ is not kept for streamwise Reynolds stress ,shown on the op row. Additionally, the merging wake is less apparent for the control case and visually the Reynolds stresses are smaller downstream when there is control. The time average velocity magnitude of the flow domain is shown on figure 32. It shows a slightly smaller velocity magnitude downstream of the prisms indicates the flow tends to be less energetic and exhibits reduced fluctuations in velocity.

On table 2, the aerodynamic coefficients is compared when there a closed-loop controller is implemented. For the case when $R = 20$, it is clear that it amplifies the cortex shedding process as the drag is increased. For $R = 60$, the controller was able to decrease the drag by 5%. The mean lift coefficient was increased by a small amount. The C'_D signal for the last 200 time units is shown on figure 31 and it is clear that the fluctuations has been damped to some degree.

In summary, the use of a linear optimal LQG controller that is appropriately tuned , is able to suppress

the effects of the vortex shedding.

Table 2: Aerodynamic coefficients with and without control

	No Control	Control ($R = 20$)	Control ($R = 60$)
$\overline{C_L}$	0.201	0.492	0.206
$\overline{C_D}$	2.035	2.224	1.932

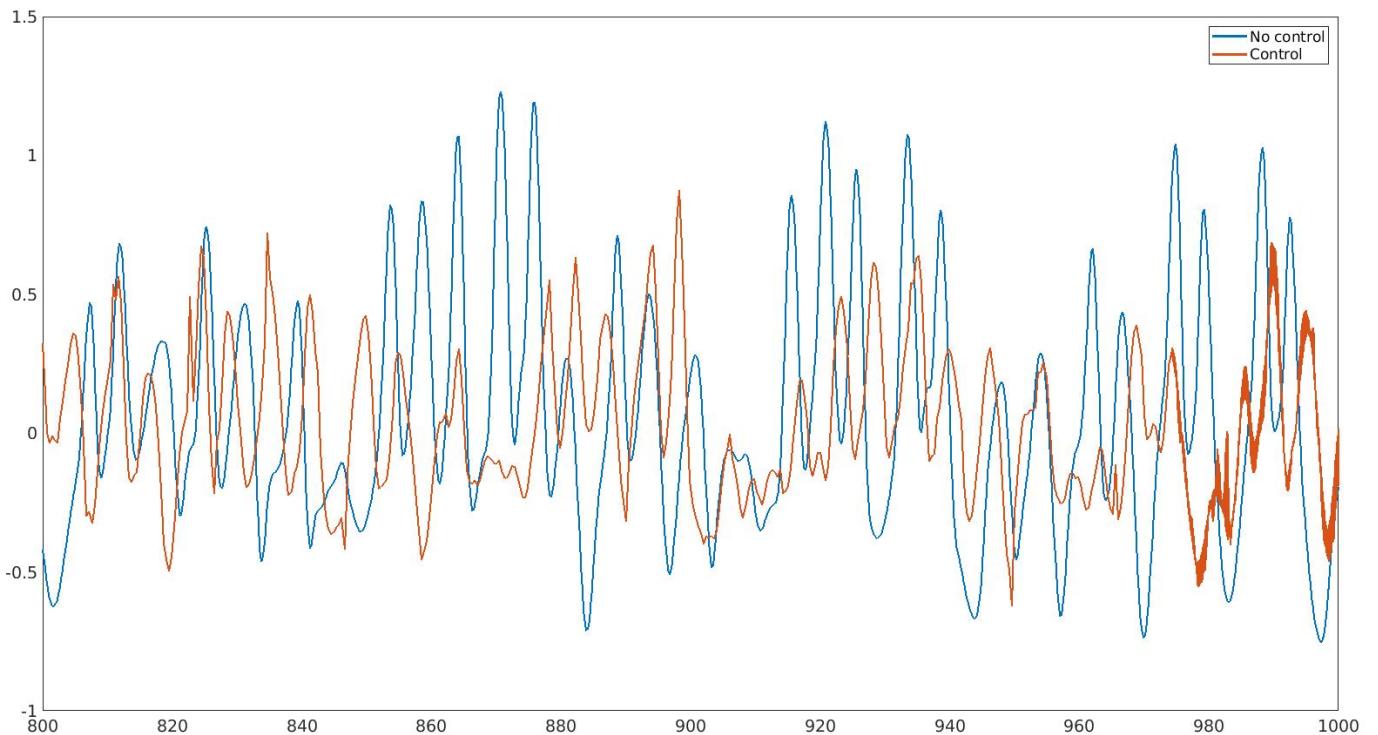


Figure 31: Fluctuation of drag coefficient with and without control

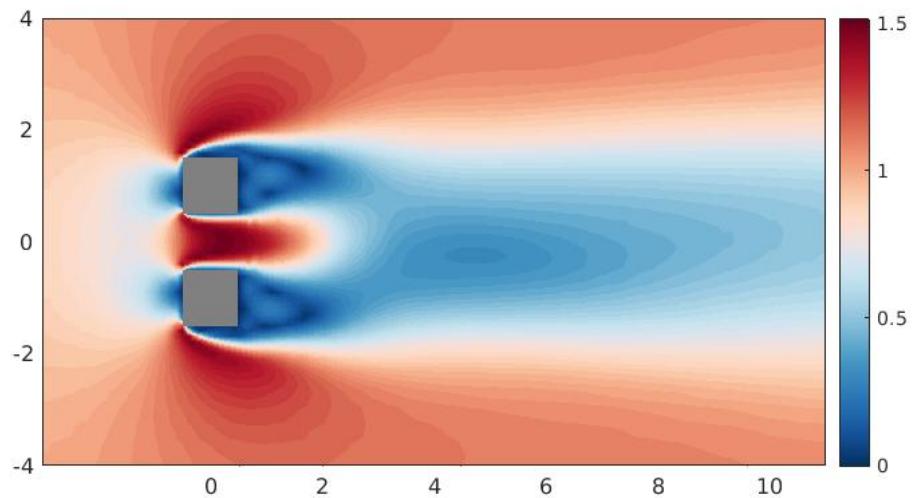
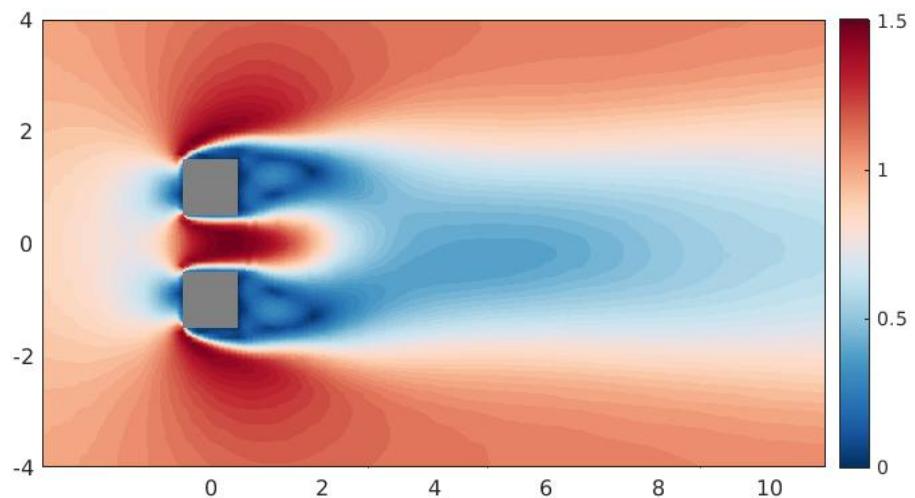


Figure 32: Time average velocity magnitude for DNS and LQG controller

6 Conclusions and future work

The aim of this project was to utilise previous studies to develop a linear controller for a highly non-linear flow to suppress the effects of vortex shedding. The chosen chaotic model was a fluid flow along two square prisms at $Re = 200$ and $g^* = 1$ and was simulated computationally through a 2D CFD environment. Spectral analysis was performed on the aerodynamic coefficients and the velocities at different locations in the wake to verify the spectrum with previous studies. Peaks were identified at 0.08 and 0.15 which corresponds to the emanating jet through the gap and the vortex shedding, respectively.

A popular optimal linear controller was chosen that was found to be effective for the control of noise amplifier and open cavity flows. The linear quadratic gaussian controller involves an estimator and a controller which can be designed separately. This involved a data-driven method by first capturing the high-dimensional flow in a computationally affordable manner through the use of a reduced order model, more specifically the proper orthogonal decomposition method via snapshots. This results in the decomposition of spatial and temporal modes which were ranked and found that 50 modes were able to capture around 97% of the fluctuation energy. The reconstruction of the flow is compared with the DNS solution as more modes are added.

System identification (**n4sid**) was carried out using only the output data (temporal modes) to identify a stochastic system, with a model order of 50, to design a Kalman filter which included the optimal sensors location found by using the greedy QR pivoting algorithm. A comparison of the sensors found from QR and sensors found from POD peaks shown in [5] verifies the superior performance of the sub-optimal sensors as the sensors were less clustered. It was revealed that only 20 sensors were able to estimate the flow to a good level for the whole domain and was justified given that more sensors were placed at the far wake region.

The controller aspect of the LQG is based on the linear quadratic regulator which aims to minimize the cost functional, where the control objective is the energy of the temporal coefficients which correspond to the POD modes. It was shown that when the controller is tuned such that $R = 60$, the mean drag coefficient was able to decrease by 5%. The spectra of the points in the wake are analyzed and found the suppression of the characteristic frequencies in the mid and far wake but there was no suppression of frequencies in the near wake just rear of the prisms. Additionally, signals of the fluctuating velocity and the fluctuating drag are plotted and it shows the fluctuations themselves are attenuated. The peak of the cross-stream root mean squared Reynolds stress was weakened by 25%.

Overall, the objective of controlling non-linear chaotic flow using a linear controller was met given the reduction in drag and the dampening of the effects from vortex shedding.

6.1 Future developments

The controller presented for this chaotic flow proves to show an exciting and different avenue for highly non-linear control. However, the extent of the LQG to control vortex shedding should be explored. Firstly, the control scenarios explored within this work did not try a bigger range of values for the input cost weighted matrix. The given matrix $R = 60$, was found to give desirable results but the maximum magnitude of the actuation velocity was 0.5 m/s. In [41], the use of a non-linear controller was able to completely suppress vortex shedding past a circular cylinder with a root mean squared actuation of 8 % of the free-stream velocity and therefore a greater R should be attempted. Moreover, like the sensor locations, the optimal actuator locations and number of actuators should be explored where [42] concludes that the actuators should be able to control the near and far wake areas. In the event that a linear controller is able to control vortex shedding to a level similar to non-linear controllers, the next step would be to explore the cost associated with each controller. Future research should be performed on the above mentioned topics.

References

- [1] S.B. Pope. *Turbulent Flows*. Cambridge University Press, 2000.
- [2] J.L. Lumley. *The Structure of Inhomogeneous Turbulent Flows*. 1967.
- [3] C.W Rowley and S.T.M Dawson. “Model Reduction for Flow Analysis and Control”. In: *Annual Review of Fluid Mechanics* 49.1 (2017), pp. 387–417. DOI: [10.1146/annurev-fluid-010816-060042](https://doi.org/10.1146/annurev-fluid-010816-060042).
- [4] J.N Kutz and S.L. Brunton. *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems and Contrpl*. Cambridge University Press, 2019.
- [5] F. Savarino and G. Papadakis. *Reconstruction of irregular flow dynamics around two square cylinders from sparse measurements using a data-driven algorithm*. 2022. arXiv: [2209.03290](https://arxiv.org/abs/2209.03290) [physics.flu-dyn].
- [6] J Lin. *Review of research on low-profile cortex generators to control boundary-layer separation*. Tech. rep. NASA Langley Research Center, 2002.
- [7] G Godard and M Stanislas. *Control of a decelerating boundary layer part 1: Optimization of passive vortex generators*. Tech. rep. Aerospace Science and Technology, 2006.
- [8] W Shih et al. *Experiments on flow past rough circular cylinders at large reynolds numbers*. Tech. rep. Journal of Wind Engineering and Industrial Aerodynamics, 1993.
- [9] P.W Bearman and J.K Harvey. *Control of circular cylinder flow by the use of dimples*. Tech. rep. AIAA Journals, 1993.
- [10] J Guzman-Inigo. “Estimation and control of noise amplifier flows using data-based approaches”. PhD thesis. Ecole Polytechnique, 2015.
- [11] L. N. Cattafesta and M Sheplak. “Actuators for Active Flow Control”. In: *Annual Review of Fluid Mechanics* 43 (2011), pp. 247–272. DOI: [10.1146/annurev-fluid-122109-160634](https://doi.org/10.1146/annurev-fluid-122109-160634).
- [12] P. T. Tokumaru and P. E. Dimotakis. “Rotary oscillation control of a cylinder wake”. In: *Journal of Fluid Mechanics* 224 (1991), 77–90. DOI: [10.1017/S0022112091001659](https://doi.org/10.1017/S0022112091001659).
- [13] B Protas and J Wesfreid. “Drag force in the open-loop control of the cylinder wake in the laminar regime”. In: *Physics of Fluids - PHYS FLUIDS* 14 (2002). DOI: [10.1063/1.1432695](https://doi.org/10.1063/1.1432695).
- [14] D Sipp. “Open-loop control of cavity oscillations with harmonic forcings”. In: *Journal of Fluid Mechanics* 708 (2012), 439–468. DOI: [10.1017/jfm.2012.329](https://doi.org/10.1017/jfm.2012.329).
- [15] J Kim and T Bewley. “A Linear Systems Approach to Flow Control”. In: *Annual Review of Fluid Mechanics* 39 (Dec. 2006), pp. 383–417. DOI: [10.1146/annurev.fluid.39.050905.110153](https://doi.org/10.1146/annurev.fluid.39.050905.110153).
- [16] N Gautier et al. “Closed-loop separation control using machine learning”. In: *Journal of Fluid Mechanics* 770 (May 2015), pp. 442–457. DOI: [10.1017/jfm.2015.95](https://doi.org/10.1017/jfm.2015.95).
- [17] Alberto R. and Luca M. *Data-driven prediction and control of extreme events in a chaotic flow*. 2022. arXiv: [2204.11682](https://arxiv.org/abs/2204.11682) [physics.flu-dyn].
- [18] K. Zeng, A.J. Linot, and M.D. Graham. “Data-driven control of spatiotemporal chaos with reduced-order neural ODE-based models and reinforcement learning”. In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 478.2267 (2022). DOI: [10.1098/rspa.2022.0297](https://doi.org/10.1098/rspa.2022.0297).
- [19] F. Nazarimehr et al. “Fuzzy predictive controller for chaotic flows based on continuous signals”. In: *Chaos, Solitons Fractals* 106 (2018), pp. 349–354. ISSN: 0960-0779. DOI: <https://doi.org/10.1016/j.chaos.2017.12.005>. URL: <https://www.sciencedirect.com/science/article/pii/S0960077917305076>.
- [20] S. Ma et al. “Wake of two side-by-side square cylinders at low Reynolds numbers”. In: *Physics of Fluids* 29.3 (Mar. 2017). 033604. ISSN: 1070-6631. DOI: [10.1063/1.4979134](https://doi.org/10.1063/1.4979134). eprint: https://pubs.aip.org/aip/pof/article-pdf/doi/10.1063/1.4979134/16116922/033604__1__online.pdf. URL: <https://doi.org/10.1063/1.4979134>.
- [21] M.D Graham. *Data-driven reduced order modelling and control of flows with complex chaotic dynamics*. 2022. DOI: <https://doi.org/10.52843/cassyni.r5k9ks>.
- [22] A Barbagallo, D Sipp, and P.J. Schmid. “Closed-loop control of an open cavity flow using reduced-order models”. In: *Journal of Fluid Mechanics* 641 (2009), 1–50. DOI: [10.1017/S0022112009991418](https://doi.org/10.1017/S0022112009991418).

- [23] E. Adeeb, B.A. Haider, and C.H. Sohn. “Flow interference of two side-by-side square cylinders using IB-LBM – Effect of corner radius”. In: *Results in Physics* 10 (2018), pp. 256–263. ISSN: 2211-3797. DOI: <https://doi.org/10.1016/j.rinp.2018.05.039>.
- [24] B Gera, Pavan K Sharma, and RK Singh. “CFD analysis of 2D unsteady flow around a square cylinder”. In: *International Journal of applied engineering research* 1.3 (2010), p. 602.
- [25] LSS WIKI. *StarCCM+ Directed/Structured Mesh*. 2018. URL: <https://wikis.ovgu.de/lss/doku.php?id=guide:starccm:directedmesh>.
- [26] J.A.A. Gomez. *Data-driven Flow Reconstruction around a Fish from Shear Stress or Pressure Measurements*. Tech. rep. London: Imperial College London, 2022.
- [27] ANSYS Fluent 12.0. *Theory Guide - 18.4.3 Pressure-Velocity Coupling*. URL: <https://www.afea.it/project/neptunius/docs/fluent/html/th/node373.htm>.
- [28] L. Sirovich. “Turbulence and the dynamics of coherent structures. I. Coherent structures”. In: *Quarterly of Applied Mathematics* (1987).
- [29] F. Savarino. *Reconstruction of flow around two side-by-side square cylinders using System Identification*. Tech. rep. London: Imperial College London, 2021.
- [30] P Holmes, J.L. Lumley, and G. Berkooz. *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*. Cambridge University Press, 1996.
- [31] MATLAB. *Singular value decomposition*. URL: <https://www.mathworks.com/help/matlab/ref/double.svd.html>.
- [32] J. Guzmán-Íñigo, M. A. Sodar, and G. Papadakis. “Data-based, reduced-order, dynamic estimator for reconstruction of nonlinear flows exhibiting limit-cycle oscillations”. In: *Physical Review Fluids* 4.11 (2019). DOI: [10.1103/physrevfluids.4.114703](https://doi.org/10.1103/physrevfluids.4.114703).
- [33] S.J Qin. *An overview of subspace identification*. Computers chemical engineering, 2006.
- [34] P. Van Overschee and B. De Moor. *A unifying theorem for three subspace system through identification algorithms*. Automatica, 1995.
- [35] B.D.O. Anderson and J.B. Moore. *Optimal Filtering*. Prentice-Hall, Inc, 1979.
- [36] K. Manohar, J.N. Kutz, and S.L. Brunton. *Optimal Sensor and Actuator Selection using Balanced Model Reduction*. 2020. arXiv: [1812.01574 \[math.OC\]](https://arxiv.org/abs/1812.01574).
- [37] A. C. Antoulas. *Approximation of Large-Scale Dynamical Systems*. Society for Industrial and Applied Mathematics, 2005. DOI: [10.1137/1.9780898718713](https://doi.org/10.1137/1.9780898718713).
- [38] T. Duriez, S.L. Brunton, and B.R. Noack. *Machine Learning Control – Taming Nonlinear Dynamics and Turbulence*. Springer, 2017.
- [39] A. Hervé et al. “A physics-based approach to flow control using system identification”. In: *Journal of Fluid Mechanics* 702 (2012), 26–58. DOI: [10.1017/jfm.2012.112](https://doi.org/10.1017/jfm.2012.112).
- [40] MATLAB. *What Is LQR Optimal Control? — State Space, Part 4*. URL: <https://www.mathworks.com/videos/state-space-part-4-what-is-lqr-control-1551955957637.html>.
- [41] X. Mao, H.M. Blackburn, and S.J. Sherwin. “Nonlinear optimal suppression of vortex shedding from a circular cylinder”. In: *Journal of Fluid Mechanics* 775 (2015), 241–265. DOI: [10.1017/jfm.2015.304](https://doi.org/10.1017/jfm.2015.304).
- [42] B. Jin, S.J. Illingworth, and R.D. Sandberg. “Optimal sensor and actuator placement for feedback control of vortex shedding”. In: *Journal of Fluid Mechanics* 932 (2022), A2. DOI: [10.1017/jfm.2021.948](https://doi.org/10.1017/jfm.2021.948).

A Appendix

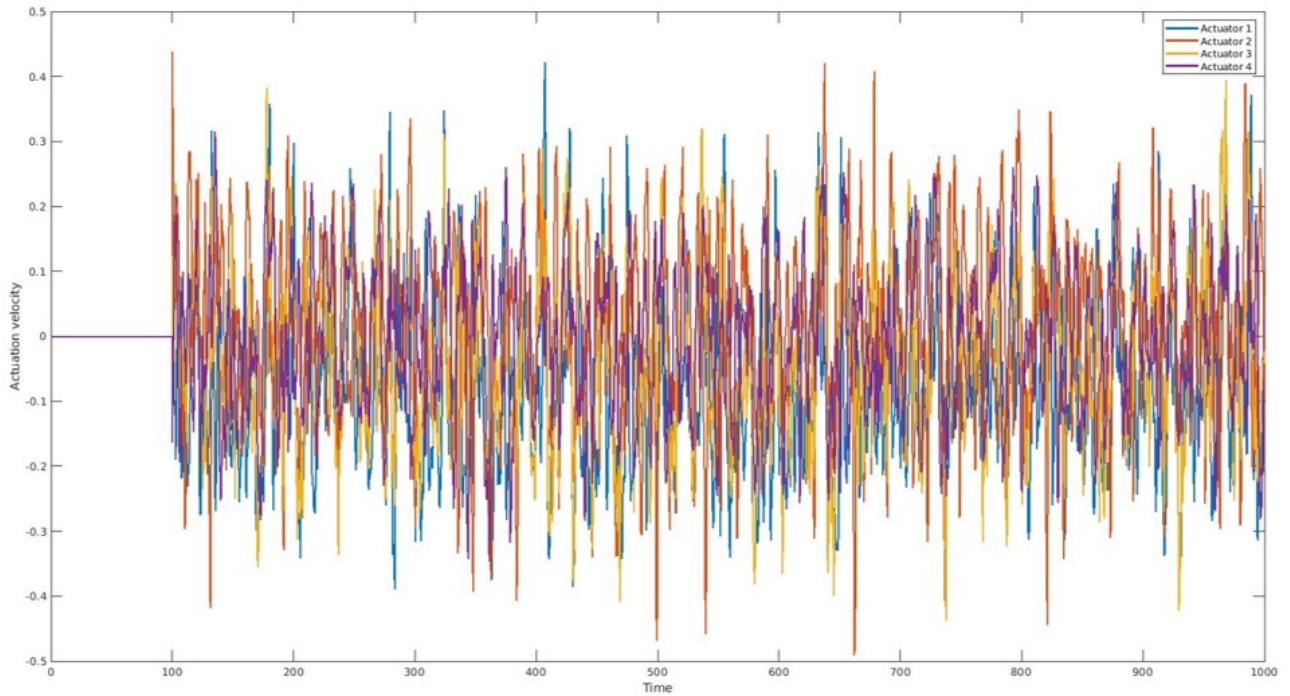


Figure 33: Signal of the actuation velocity for the 4 actuators

It can be seen on figure 33, that the actuation is no greater than 0.5 m/s, however this is much greater than the velocity of 8% of the free-stream velocity which was suggested to completely suppress vortex shedding in [41].