PROCEEDINGS A

rspa.royalsocietypublishing.org

Research

Article submitted to journal

Subject Areas:
Nonlinear control, Shadowing

Keywords:
Feedback control, Chaos, Shadowing, Kuramoto Sivashinsky Equation

Author for correspondence:
George Papadakis
e-mail: g.papadakis@imperial.ac.uk

Feedback control of chaotic systems using Multiple Shooting Shadowing and application to Kuramoto Sivashinsky equation

Karim Shawki and George Papadakis

Department of Aeronautics, Imperial College London, Exhibition Road, London, SW7 2AZ, UK

We propose an iterative method to evaluate the feedback control kernel of a chaotic system directly from the system's attractor. Such kernels are currently computed using standard linear optimal control theory, known as Linear Quadratic Regulator (LQR) theory. This is however applicable only to linear systems, which are obtained by linearising the system governing equations around a target state. In the present paper, we employ the Preconditioned Multiple Shooting Shadowing (PMSS) algorithm to compute the kernel directly from the non-linear dynamics, thereby bypassing the linear approximation. Using the adjoint version of the PMSS algorithm, we show that we can compute the kernel at any point of the domain in a single computation. The algorithm replaces the standard adjoint equation (that is ill-conditioned for chaotic systems) with a well-conditioned adjoint, producing reliable sensitivities which are used to evaluate the feedback matrix elements. We apply the idea to the Kuramoto Sivashinsky equation. We compare the computed kernel with that produced by the standard LQR algorithm and note similarities and differences. Both kernels are stabilising, have compact support and similar shape. We explain the shape using two-point spatial correlations that capture the streaky structure of the solution of the uncontrolled system.

© The Authors. Published by the Royal Society under the terms of the Creative Commons Attribution License http://creativecommons.org/licenses/by/4.0/, which permits unrestricted use, provided the original author and source are credited.

THE ROYAL SOCIETY PUBLISHING

1. Introduction

Control of chaotic systems is a very active field of research. The objective is to find the actuation (control input) that will make the system meet a desired objective. Common examples, from the area of turbulence, include drag reduction, transition delay, suppression of recirculation zones in the suction side of airfoils etc. The control problem is usually formulated mathematically as an optimisation problem (usually minimisation of a functional) and is solved using standard techniques from the calculus of variations [1]. The solution method is iterative and relies on forward (in time) integration of the governing equations followed by backward (reverse in time) integration of the adjoint equations (see [2] for an application to control of transitional flow). The adjoint equations are a set of linearised equations around the current trajectory of the system in phase space. They are obtained when applying the optimality conditions to the augmented functional and using integration by parts (refer to [3] for more details). From the adjoint variables, the sensitivity of the functional to the control input(s) is obtained which is used to update the latter in the next iteration.

For linear time-invariant systems with a quadratic control objective, application of the aforementioned calculus of variation technique results in a set of three equations that can be reduced to just one Riccati equation for a matrix P using the sweep method [4]. This matrix relates linearly the adjoint and state variables of the system. From P, the feedback matrix K, that relates the optimal input to the states of the system, can be easily obtained. This is a very well developed theory, known as Linear Quadratic Regulator (LQR) optimal control theory [5]. It has been extended to handle noisy measurements and unknown system dynamics, modelled either as white Gaussian noise or as maximally malevolent; these are known as LQG and H_∞ (or robust) approaches respectively [5]. In the area of flow control, all approaches have been applied successfully to suppress disturbances described by the linearised Navier-Stokes equations to a variety of flow settings [6, 7, 8, 9]. Application of linear controllers is also effective for systems with energy-conserving non-linearities, for example channel flows. For such cases, in the energy equation, the integral of the non-linear terms over the domain vanishes. The underlying physical meaning is that the role of non-linearity is to transfer energy from larger to smaller scales (a process known as energy cascade) but do not generate energy. The non-linear terms are therefore passive and this has motivated the application of passivity-based optimal linear control to suppress turbulence in channel flows at low Reynolds number, $Re_\tau = 100$, see [10, 11].

The wider application of model-based, linear feedback control strategies to fluid problems is however restricted by the very high system dimension, N, which makes the solution of the Riccati equation to extract P intractable (the cost scales as $O(N^3)$). This problem arises for complex geometries that do not possess one or more homogeneous directions. In such cases, the standard practise is to derive a reduced order model for the uncontrolled flow, synthesise the controller based on this model, and then apply it to the full system (see [9] for application of this approach to suppress linear 3D disturbances in a boundary layer flow). This approach however is not optimal as the reduced order model captures the dynamics of the uncontrolled flow (open loop) and not that of the controlled flow (closed loop).

A few attempts have been made to bypass the solution of the Riccati equation, and therefore the need for a reduced order model. These approaches compute directly the elements of the feedback matrix K using iterative methods that rely on the integration of the governing and adjoint equations in a forward/backward loop, exactly as described in the beginning of this section. They scale to large N and hence are suitable for systems arising from the discretisation of the Navier-Stokes equations in complex domains. In [12, 13] the authors have proposed a method that can provide the elements of the matrix row-by-row. It is based on the repeated iterative computation of the adjoint of a forward problem, the latter defined for the direct-adjoint vector pair associated with the LQR problem. The method was extended to the estimation problem in [14] and applied successfully to a 2D boundary layer, while in [15] the method was extended to robust (H_∞) control. In [16, 17], all matrix elements are computed directly. The cost

function of the LQR problem is written in terms of K and the latter is computed iteratively using forward/backward marching of the governing and adjoint equations.

All the above approaches have been applied to linear, time-invariant systems. In the present paper, we use instead the non-linear governing equations as constraints to compute K directly. We extract this matrix from the system attractor and do not apply any model reduction (an area which is far less well developed for non-linear systems compared to linear ones). The application of the iterative approach however to extract K is not straightforward. The reason is that for chaotic systems (which are always non-linear), during the backward integration step, the adjoint variables grow exponentially at a rate $O(e^{\lambda_{max}t})$, where λ_{max} is the maximum Lyapunov exponent of the system. A different methodology is therefore required to compute the adjoint variables that quantify the sensitivity of the cost function to the elements of K.

Different approaches have been proposed to deal with this problem [18, 19]. The Least Squares Shadowing (LSS) method [18] is the most promising among them. The method finds a nearby trajectory $\mathbf{u}'(\tau)$ that *shadows* the reference trajectory $\mathbf{u}_{ref}(t)$ of the chaotic system. The difference $|\mathbf{u}'(\tau) - \mathbf{u}_{ref}(t)|$ remains bounded in time, so accurate sensitivities can be computed. The original LSS algorithm is computationally expensive for large systems, and the Multiple Shooting Shadowing (MSS) [20] and the non-intrusive (NILSS) [21] variants are both faster and less memory demanding. A preconditioner was recently proposed by the authors to accelerate the convergence rate of the MSS variant [22]. When the sensitivity of a given objective to a large number of control variables is required, as in our case, adjoint methods offer the most efficient approach [1], since the cost of computation is independent of the number of variables. Adjoint versions of LSS and its variants have also been proposed in the literature [18, 20, 23].

In this paper we employ the Preconditioned Multiple Shooting Shadowing (PMSS) method to compute the adjoint variables and from them, in a single computation, all the elements of the optimal feedback gain matrix. We apply this idea to the Kuramoto Sivashinsky equation (KSE) with Dirichlet boundary conditions, that result in an ergodic system [24]. There is also an important additional benefit. For this type of boundary conditions, the non-linearity of the system is energy conserving, a property which makes standard methods, like LQR, effective. Thus we can compare the control kernels produced by the shadowing and LQR approaches. We note that very few attempts have been made to solve an optimisation problem using LSS before [25, 26], and none have considered the computation of the feedback matrix.

The feedback control of the KSE (with periodic boundary conditions) has been considered in the past. In [27, 28] the authors discretised the equation by applying a Galerkin projection to the eigenmodes of the linear operator and then pole placement to stabilise its dynamics. The (linearly) unstable modes were separated from the stable ones and a controller was synthesised for the former. The number of actuators was based on the number of unstable modes. In [29, 30], the optimal control problem considered the cost of actuation and the iterative algorithm also found the optimal actuator locations that minimise the distance between the solution and a desired state (which can be a travelling wave) over a given time period T. Recently, 'model-free' approaches, like deep reinforcement learning (DRL), were proposed [31]. This is a data-driven method that finds the optimal control laws using only local measurements.

The structure of this paper is as follows: Section (2) summarises the control problem and the two algorithms. In Section (3) we apply both control strategies to the KSE and compare the control kernels, while in Section (4) we summarise the performance of the shadowing algorithm. We conclude in Section (5).

2. Formulation of the control problem

We consider a general dynamical system governed by a set of ordinary differential equations (ODEs) in general form

$$\frac{d\mathbf{u}}{dt} = \mathbf{f}(\mathbf{u}) \tag{2.1}$$

where $\mathbf{u}$ is the state vector of length N. If necessary, we assume that an appropriate spatial discretisation method (for example finite difference, finite volume etc.) has been applied to convert a system of partial differential equations to the ODE set (2.1). In this case, $\mathbf{u}$ represents the vector of discrete values of the corresponding continuous variables.

The above set describes mathematically the evolution of the uncontrolled system. For control purposes, we seek an actuation $\mathbf{s}(t)$ that will modify the behaviour of the system to meet a desired objective. Assuming M actuators (in which case $\mathbf{s}(t)$ becomes a vector of length M), the controlled system takes the form

$$\frac{d\mathbf{u}}{dt} = \mathbf{f}(\mathbf{u}) + B\mathbf{s}(t) \tag{2.2}$$

where B is the input matrix (size $N \times M$) that determines the spatial distribution of the actuation. Moreover, we assume that the actuation $\mathbf{s}(t)$ takes the linear feedback control form

$$\mathbf{s}(t) = -K\left(\mathbf{u}(t) - \mathbf{u}_{targ}\right) \tag{2.3}$$

where K is an $M \times N$ matrix, and $\mathbf{u}_{targ}$ is a desired (target) state of the system. The controlled system then becomes

$$\frac{d\mathbf{u}}{dt} = \mathbf{f}(\mathbf{u}) - BK\mathbf{u} \equiv \mathbf{h}(\mathbf{u}, K), \tag{2.4}$$

where for future reference we have defined $\mathbf{h}(\mathbf{u}, K) = \mathbf{f}(\mathbf{u}) - BK\mathbf{u}$. The objective of this paper is to compare the matrix K obtained using the standard linear optimal control theory (also known as LQR) and the one obtained from Preconditioned Multiple Shooting Shadowing (PMSS). We briefly describe both approaches below.

(a) Control using Linear Quadratic Regulator (LQR)

The LQR theory is developed for linear time-invariant (LTI) systems. We first linearise the governing ODE set around a stationary target state, $\mathbf{u}_{targ}$, i.e. we substitute $\mathbf{u} = \mathbf{u}_{targ} + \mathbf{u}'$ into (2.1), and keeping only the linear terms, we arrive at

$$\frac{d\mathbf{u}'}{dt} = A_l\mathbf{u}' \tag{2.5}$$

where $A_l = \frac{\partial \mathbf{f}}{\partial \mathbf{u}}|_{u_{targ}}$ is the Jacobian matrix evaluated at the target sate. Actuation is then applied to the linearised system, i.e.

$$\frac{d\mathbf{u}'}{dt} = A_l\mathbf{u}' + B\mathbf{s}(t) \tag{2.6}$$

LQR finds the optimum $\mathbf{s}(t)$ which minimises the following objective:

$$J = \int_0^\infty \left(\mathbf{u}'^T Q\mathbf{u}' + \mathbf{s}^T R\mathbf{s}\right) dt \tag{2.7}$$

subject to (2.6), where Q and R are weighting matrices (size $N \times N$ and $M \times M$ respectively). Both are symmetric, $Q \geq 0$ and $R > 0$ (i.e. positive semi-definite and definite respectively). The solution to the optimisation problem results in an optimum actuation $\mathbf{s}(t)$ which is related to the state $\mathbf{u}(t)$ via equation (2.3) and the feedback matrix K is obtained from

$$K = R^{-1}(B^T P) \tag{2.8}$$

where matrix P (size $N \times N$) is the solution to the algebraic Riccati equation:

$$A_l^T P + PA_l - PBR^{-1}B^T P + Q = 0 \tag{2.9}$$

The cost of the solution of this equation scales as $O(N^3)$. In the present work, we set $B = I$ and the weighting matrices $Q = I$ and $R = cI$, where c is a positive constant. The optimal feedback

controller is then given by

$$K = \frac{1}{c} P \tag{2.10}$$

The derivation of equations (2.8) and (2.9) can be found in standard control textbooks [4, 32]. The optimum feedback controller (2.8) is then applied to the full-non linear system (2.4). We note that for an LTI system, the linear relationship (2.3) between actuation and state is exact, and is a direct outcome of the solution of the optimisation problem. In the present work, we set the target state $\mathbf{u}_{targ} = \mathbf{0}$, i.e. we want the actuation to bring the system to rest.

(b) Control using Preconditioned Multiple Shooting Shadowing (PMSS)

We seek the control matrix K that solves the non-linear control problem

$$\underset{K}{\text{Minimise}} \quad \overline{J^{(\infty)}}(\mathbf{u}, K) = \lim_{T\to\infty} \frac{1}{T} \int_0^T J(\mathbf{u}, K)\, dt \tag{2.11a}$$

$$\text{subject to} \qquad \frac{d\mathbf{u}}{dt} = \mathbf{f}(\mathbf{u}) - BK\mathbf{u} \left(= \mathbf{h}(\mathbf{u}, K)\right) \tag{2.11b}$$

where in analogy with (2.7) we set

$$J(\mathbf{u}, K) = \mathbf{u}^T Q \mathbf{u} + \mathbf{s}^T R \mathbf{s} = \mathbf{u}^T Q \mathbf{u} + (K\mathbf{u})^T R\, (K\mathbf{u}) \tag{2.12}$$

Note that in (2.12), the full state $\mathbf{u}$ appears, not the perturbation around the target, $\mathbf{u}'$, as in (2.5). Since however we have selected $\mathbf{u}_{targ} = \mathbf{0}$, both control problems are aiming to bring the system to rest. We employ $\overline{J^{(\infty)}}$ to indicate averaging over an infinitely long period. For averaging over finite time T, we use

$$\overline{J^{(T)}} = \frac{1}{T} \int_0^T J(\mathbf{u}, K)\, dt \tag{2.13}$$

instead.

The optimisation problem (2.11) can be solved iteratively using the sensitivities $\frac{d\overline{J^{(\infty)}}}{dK}$ and coupling with an updating method, for example gradient descent,

$$K^{(i+1)} = K^{(i)} - a^{(i)} \left(\frac{d\overline{J^{(\infty)}}}{dK} \right)^{(i)} \tag{2.14}$$

where i is the iteration number and $a^{(i)}$ is a step size. We explain how to evaluate the sensitivities for all elements of the matrix K (in total N^2 values) with a single computation using the adjoint version of PMSS in the following subsection. The steps of the iterative algorithm are summarised below.

Control algorithm:
Inputs: T, $\mathbf{u}_0$, $K^{(0)}$, ϵ. Output: K (converged control matrix)

(i) Set $i = 0$
(ii) Integrate $\dot{\mathbf{u}} - \mathbf{h}(\mathbf{u}, K^{(0)}) = 0$ with initial condition $\mathbf{u}(0) = \mathbf{u}_0$ in the interval $[0, T]$ to obtain $\mathbf{u}^{(0)}(t)$. Compute $\overline{J^{(T)}}^{(0)}$.
(iii) Call **PMSS solver** to obtain $\left(\frac{d\overline{J^{(T)}}}{dK} \right)^{(i)}$ for the trajectory $\mathbf{u}^{(i)}(t)$
(iv) Compute $a^{(i)}$ using an appropriate algorithm (for example backtracking or Brent's method [33])
(v) Update the control parameters: $K^{(i+1)} = K^{(i)} - a^{(i)} \left(\frac{d\overline{J^{(T)}}}{dK} \right)^{(i)}$
(vi) Integrate $\dot{\mathbf{u}} - \mathbf{h}(\mathbf{u}, K^{(i+1)}) = 0$ and store $\mathbf{u}^{(i+1)}(t)$. Compute $\overline{J^{(T)}}^{(i+1)}$. (Note: This step is included within step 4 if backtracking is employed)

(vii) If $\left|\frac{\overline{J^{(T)}}^{(i+1)}-\overline{J^{(T)}}^{(i)}}{\overline{J^{(T)}}^{(i)}}\right| < \epsilon$ break, and output $K = K^{(i+1)}$. If not, set $i = i+1$ and return to step (iii).

Step 4 of the algorithm typically requires multiple evaluations of (2.11b) to compute $a^{(i)}$. In this paper we use backtracking, which guarantees monotonic reduction of $\overline{J^{(T)}}^{(i)}$, and is considerably less expensive than other extrema finding methods (such as Brent's method). Backtracking requires the computation of $\mathbf{u}(t)$ and $\overline{J^{(T)}}$, so these are obtained on Step 4 (and make step 6 redundant). The next subsection describes in detail the computation of the sensitivities (step 3).

The above algorithm requires $\overline{J^{(T)}}$ to be ergodic (i.e. independent of the initial condition), as will be explained in Section (2i). It must also be noted that the cost function $J(\mathbf{u}, K)$, and therefore $\overline{J^{(T)}}(\mathbf{u}, K)$, are generally non-convex with respect to the elements of K [34]. Therefore in general, convergence to the global minimum is not guaranteed.

(i) Computation of sensitivities $\frac{d\overline{J^{\infty}}}{dK}$ using an adjoint algorithm

We are seeking the derivative of one objective, $\overline{J^{(\infty)}}$ (in practise $\overline{J^{(T)}}$), to multiple inputs (the elements of matrix K). For such cases, adjoint algorithms offer an ideal solution, because their cost is independent from the number of matrix elements.

We start by explaining how to compute the derivative to a single parameter, say β, evaluated at $\beta = \beta_0$. i.e. we seek $\frac{d\overline{J^{(\infty)}}}{d\beta}$ under the non-linear constraint,

$$\frac{d\mathbf{u}}{dt} = \mathbf{h}(\mathbf{u}, \beta = \beta_0) \tag{2.15}$$

The linear *tangent* equation,

$$\frac{d\mathbf{v}}{dt} = \left.\frac{\partial \mathbf{h}}{\partial \mathbf{u}}\right|_{\mathbf{u},\beta_0} \mathbf{v} + \left.\frac{\partial \mathbf{h}}{\partial \beta}\right|_{\mathbf{u},\beta_0} \tag{2.16}$$

where $\left.\frac{\partial \mathbf{h}}{\partial \mathbf{u}}\right|_{\mathbf{u},\beta_0}$ is the Jacobian, governs the evolution of $\mathbf{v}(t) = \frac{d\mathbf{u}(t)}{d\beta}$ with initial condition $\mathbf{v}(0) = \frac{d\mathbf{u}(0)}{d\beta}$. This equation is obtained by perturbing β around the reference value $\beta = \beta_0$ and linearising (2.15) around the β_0-trajectory in phase space. The variable $\mathbf{v}(t)$ can then be used to compute the sensitivity $\frac{d\overline{J^{(\infty)}}}{d\beta}$ by evaluating an integral obtained after application of chain rule to (2.11a). The solution $\mathbf{v}(t)$ of the tangent equation is however unbounded for chaotic systems (grows exponentially in time due to the divergence of nearby trajectories). Least Squares Shadowing (LSS) [18] finds a solution to (2.16) that always remains bounded in time, which means that the perturbed trajectory $\mathbf{u}'(\tau, \beta_0 + \delta\beta)$ shadows (i.e. remains close to) the reference trajectory $\mathbf{u}_{ref}(t, \beta_0)$. This is achieved by relaxing the initial condition $\mathbf{v}(0)$, and minimising the norm $\|\mathbf{v}(t)\|_2^2$, where $\mathbf{v}(t)$ is now defined as $\mathbf{v}(t) = \lim_{\delta\beta\to 0} \frac{\mathbf{u}'(\tau,\beta_0+\delta\beta)-\mathbf{u}_{ref}(t,\beta_0)}{\delta\beta}$. For ergodic systems, the sensitivity does not depend on the initial conditions, so this relaxation does not affect the result. Multiple Shooting Shadowing (MSS) [20] is a variant of LSS that minimises $\|\mathbf{v}\|_2^2$ at $P+1$ discrete checkpoints in time, that define P time segments $\left[t_{p-1}, t_p\right]$, $p = 1 \dots P$, with size $\Delta T = \frac{T}{P}$ (equal for all segments). Below we provide only a summary of MSS. For full explanation and derivation details, the reader is referred to [20]. The minimisation problem can be written as

$$\underset{\mathbf{v}_p}{\text{Minimise}} \quad \frac{1}{2}\sum_{p=0}^{P} \|\mathbf{v}_p\|_2^2 \tag{2.17a}$$

$$\text{subject to} \quad A\underline{\mathbf{v}} = \underline{\mathbf{b}}. \tag{2.17b}$$

Equation (2.17b) imposes the continuity of $\mathbf{v}$ between consecutive time segments. The size of matrix A is $NP \times N(P+1)$, while $\underline{\mathbf{v}} = \begin{bmatrix}\mathbf{v}_0^T & \mathbf{v}_1^T & \dots & \mathbf{v}_P^T\end{bmatrix}^T$ and $\underline{\mathbf{b}} = \begin{bmatrix}\mathbf{b}_1^T & \mathbf{b}_2^T & \dots & \mathbf{b}_P^T\end{bmatrix}^T$

are vectors of length $N(P+1)$ and NP, respectively. Matrix A and vector $\underline{\mathbf{b}}$ can be computed from the state transition matrix of (2.16) that satisfies,

$$\frac{d\phi^{t,\tau}}{d\tau} = \left.\frac{\partial \mathbf{h}}{\partial \mathbf{u}}\right|_{\mathbf{u},\beta_0} \phi^{t,\tau} \tag{2.18}$$

The solution to the minimisation problem (2.17) is found by introducing a set of discrete adjoint variables $\underline{\mathbf{w}} = \begin{bmatrix}\mathbf{w}_1^T & \mathbf{w}_2^T & \dots & \mathbf{w}_P^T\end{bmatrix}^T$ and deriving the Karush–Kuhn–Tucker (KKT) system

$$\begin{bmatrix} -I & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \underline{\mathbf{v}} \\ \underline{\mathbf{w}} \end{bmatrix} = \begin{bmatrix} 0 \\ \underline{\mathbf{b}} \end{bmatrix} \tag{2.19}$$

From the computational point of view, it is more convenient to solve a linear system that involves the Schur complement matrix, $S = AA^T$ with size $NP \times NP$,

$$S\underline{\mathbf{w}} = \underline{\mathbf{b}} \tag{2.20}$$

Matrix S is block tri-diagonal and symmetric positive definite, i.e. all eigenvalues are real and positive. Equation (2.20) is solved iteratively using a linear 'matrix-free' solver i.e. S is not stored; only the matrix-vector products $S\underline{\mathbf{w}}$ are required. The GMRES solver is used in this paper. Once $\underline{\mathbf{w}}$ is computed, it is substituted into (2.19) to obtain $\underline{\mathbf{v}}$, and $\frac{d\overline{J^{(T)}}}{d\beta}$ can be evaluated.

The above analysis is not useful when seeking the derivative of $\overline{J^{(T)}}$ to multiple control parameters because (2.20) would need to be solved with a different right hand side for each parameter. The adjoint version of (2.19)

$$\begin{bmatrix} -I & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \underline{\hat{\mathbf{v}}} \\ \underline{\hat{\mathbf{w}}} \end{bmatrix} = \begin{bmatrix} -\underline{\mathbf{g}} \\ 0 \end{bmatrix} \tag{2.21}$$

can used to compute the derivative of $\overline{J^{(T)}}$ to all control parameters (in our case, the elements of K) with a single solution to (2.21). In the above equation, $\underline{\hat{\mathbf{v}}}$ and $\underline{\hat{\mathbf{w}}}$ are the discrete adjoint variables of $\underline{\mathbf{v}}$ and $\underline{\mathbf{w}}$ respectively. Vector $\underline{\mathbf{g}} = \begin{bmatrix}\mathbf{g}_1^T & \mathbf{g}_2^T & \dots & \mathbf{g}_P^T & 0\end{bmatrix}^T$, where

$$\mathbf{g}_p = \frac{1}{T}\int_{t_{p-1}}^{t_p} \left.\frac{\partial J}{\partial \mathbf{u}}\right|^T \phi^{t_{p-1},t}\, dt + \frac{1}{T}(\overline{J^{(T)}} - J_p)\frac{\mathbf{h}_p\big|^T \phi^{t_{p-1},t_p}}{\|\mathbf{h}_p\|_2^2}, \tag{2.22}$$

$J_p = J(t_p)$ and $\mathbf{h}_p = \mathbf{h}(t_p)$. The Schur complement of (2.21) is

$$S\underline{\hat{\mathbf{w}}} = -A\underline{\mathbf{g}} \tag{2.23}$$

which can be solved iteratively for $\underline{\hat{\mathbf{w}}} = \begin{bmatrix}\hat{\mathbf{w}}_1^T & \hat{\mathbf{w}}_2^T & \dots & \hat{\mathbf{w}}_P^T\end{bmatrix}^T$. Notice that the standard KKT system (2.19) and its Schur complement (2.20) have identical matrices to the adjoint counterparts (2.21) and (2.23), but different right hand sides. Within each time segment, the continuous adjoint variables $\hat{\mathbf{w}}(t) = \begin{bmatrix}\hat{w}_1(t) & \hat{w}_2(t) & \dots & \hat{w}_N(t)\end{bmatrix}^T$ can be found by integrating

$$\frac{d\hat{\mathbf{w}}}{dt} = -\frac{\partial \mathbf{h}}{\partial \mathbf{u}}^T \hat{\mathbf{w}} - \frac{1}{T}\frac{\partial J}{\partial \mathbf{u}} \qquad t_{p-1} \le t < t_p \tag{2.24}$$

backwards in time in all segments with terminal conditions

$$\hat{\mathbf{w}}(t_p) = P_{t_p}\hat{\mathbf{w}}_p + \frac{1}{T}\frac{\overline{J^{(T)}} - J_p}{\mathbf{h}_p^T\mathbf{h}_p}\mathbf{h}_p \tag{2.25}$$

where $\hat{\mathbf{w}}_p$ is obtained from the solution to (2.23) and P_{t_p} is a projection operator evaluated at t_p:

$$P_{t_p} = I - \frac{\mathbf{h}_p\mathbf{h}_p^T}{\mathbf{h}_p^T\mathbf{h}_p} \tag{2.26}$$

Finally, using $\hat{\mathbf{w}}(t)$, the sensitivity of $\overline{J^{(T)}}$ to any element $K_{i,j}$ can be found by evaluating the integral

$$\frac{d\overline{J^{(T)}}}{dK_{i,j}} = \left(\sum_{p=1}^{P} \int_{t_{p-1}}^{t_p} \frac{\partial \mathbf{h}}{\partial K_{i,j}}^T \hat{\mathbf{w}}(t)\, dt \right) + \frac{1}{T} \int_0^T \frac{\partial J}{\partial K_{i,j}}\, dt, \tag{2.27}$$

where in the second term of the right hand side, J is given by (2.12). Since K in this paper is size $N \times N$, (2.27) is evaluated algebraically N^2 times to compute the full sensitivity matrix.

The most time consuming part of the algorithm is the solution of the linear system (2.23). It is well known that systems with matrices that have tightly clustered eignevalue spectra converge fast when solved iteratively [35]. For MSS, the eigenvalues of S, $\mu(S)$, are usually widely spread between very small and very large values, leading to very slow convergence. A useful measure to estimate the convergence rate is the matrix condition number, $\kappa = \frac{\mu_{max}(S)}{\mu_{min}(S)}$. For fast convergence, κ should be as close as possible to unity. In order to reduce κ, we apply a block diagonal preconditioner $\mathbf{M_{BD}}_{(l)}^{(q)}$ (refer to [22] for details) to system (2.23) that annihilates the (l) fastest growing singular modes, corresponding to the (l) largest Lyapunov exponents in each segment. The superscript (q) refers to the number of Lanczos bidiagonalisation iterations [36]. The MATLAB function 'svds', based on this algorithm, is used to compute the singular vectors that are employed to construct $\mathbf{M_{BD}}_{(l)}^{(q)}$. To account for very small eigenvalues, $0 < \mu(S) \ll 1$, Tikhonov regularisation is also used to shift the entire eigenvalue spectrum by γ. The preconditioned and regularised system takes the form

$$\left(\gamma I + \mathbf{M_{BD}}_{(l)}^{(q)} S\right) \underline{\hat{\mathbf{w}}} = -\mathbf{M_{BD}}_{(l)}^{(q)} (A\underline{\mathbf{g}}) \tag{2.28}$$

It was shown in [22] that this system has very good convergence properties, and therefore we solve (2.28) instead of (2.23). Note the enormous advantage of the adjoint approach. The system (2.28) must be solved only once in order to compute the sensitivity of $\overline{J^{(T)}}$ with respect to each element of matrix K, in total $N \times N$ values. Summarising, we have the following algorithm,

PMSS solver:
Inputs: T, P, l, q, γ. Output: $\frac{d\overline{J^{(T)}}}{dK}$

(i) For $p = 1 : P$, compute g_p and $\phi_p g_p$. Form the vector $A\underline{\mathbf{g}}$. [i]
(ii) Compute the preconditioner $\mathbf{M_{BD}}_{(l)}^{(q)}$ [ii]
(iii) Form the RHS of (2.28), $-\mathbf{M_{BD}}_{(l)}^{(q)} (A\underline{\mathbf{g}})$
(iv) Solve (2.28) for $\underline{\hat{\mathbf{w}}}$ using GMRES
(v) Integrate (2.24) backwards in time (for all segments $p = 1 : P$) with the terminal conditions in (2.25) to find $\hat{\mathbf{w}}(t)$
(vi) Compute $\frac{d\overline{J^{(T)}}}{dK}$ using (2.27)

3. Control of the Kuramoto Sivashinsky Equation

In this section we apply the two control algorithms outlined above to the standard Kuramoto-Sivashinsky equation (KSE),

$$\begin{aligned} &\frac{\partial u}{\partial t} = -u\frac{\partial u}{\partial x} - \frac{\partial^2 u}{\partial x^2} - \frac{\partial^4 u}{\partial x^4} \quad x \in [0, L] \\ &u(0,t) = u(L,t) = 0 \\ &\left.\frac{\partial u}{\partial x}\right|_{x=0} = \left.\frac{\partial u}{\partial x}\right|_{x=L} = 0 \end{aligned} \tag{3.1}$$

[i] Refer to 'Adjoint MSS Solver' in Section (5.1) of [20] for the computational details
[ii] Refer to Section (3.2) in [22] for the computational details

We chose Dirichlet and Neumann boundary conditions (instead of standard periodic conditions) so that the system is ergodic [24]. For all simulations $L = 128$, which is large enough to ensure a chaotic behaviour. The spatial derivatives are discretised using $N + 2$ equally–spaced nodes, using second order finite difference approximations as in [24]. N nodes are inside the domain and 2 are located on the boundaries. Two values of node spacing $\delta x = \frac{L}{(N+1)}$ are considered, 1 and 0.5. The variable step Runge–Kutta method (ode45 in MATLAB) was used for the temporal integration. The initial condition at $t = 0$ is obtained from a precursor integration in $-1000 \leq t \leq 0$; this ensures that the trajectory has reached the chaotic attractor at $t = 0$.

Figure (1) shows the typical streaky behaviour exhibited by the KSE in the $x - t$ plane. It is clear that there is a characteristic average streak spacing, l_{str}, with wavenumber $k = \frac{2\pi}{l_{str}}$. As will be shown later, l_{str} plays an important role in the analysis of control kernels. If the boundary conditions are periodic, the spectral energy peaks at k. This characteristic value is close to the wavenumber that maximises the eigenvalue of the linearised KSE around the rest state. For periodic boundary conditions, this corresponds to a streak spacing that can be expressed analytically as $l_{str}^{per} = 2\pi\sqrt{2}$ [37], which is equal to $l_{str}^{per} = 8.9$. As will be seen later for system (3.1), $l_{str} \approx 8.5$, therefore the choice of the boundary conditions does not affect much the streak spacing.

The time-average and root mean square (rms) of u as a function of x, $\langle \tilde{u} \rangle$ and $\tilde{u}_{RMS}$ respectively, are shown in Figure (2). The angled brackets $\langle . \rangle$ denote time-averaging, while $\tilde{.}$ represents averaging over multiple initial conditions $u_0 = u(x, 0)$. The Dirichlet boundary conditions result in sharp gradients at both ends of the domain, especially for $\tilde{u}_{RMS}$. In the middle of the domain the variation of both mean and rms is smooth.

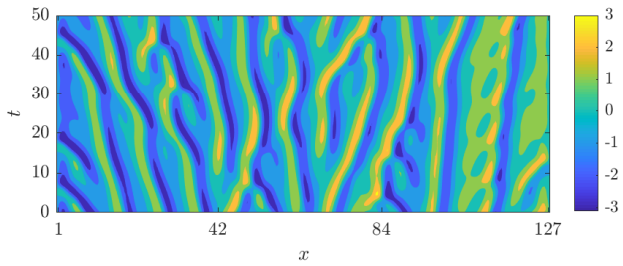


Figure 1: Contour plot of a typical solution $u(x, t)$ of (3.1).

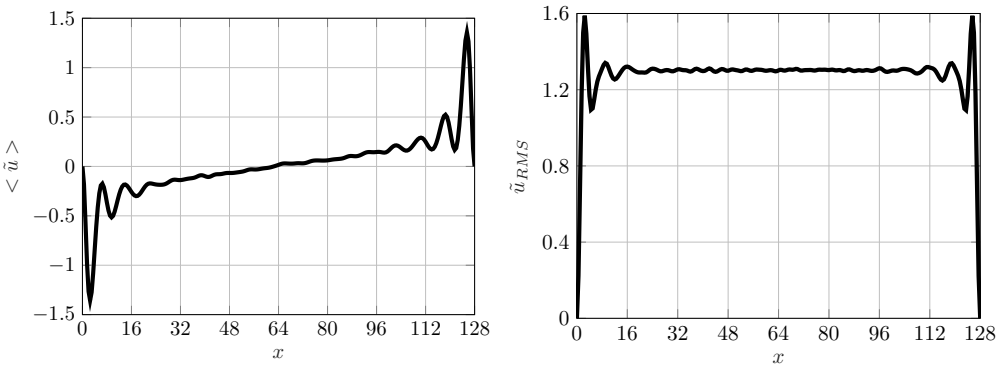
The discretised KSE equation is written as

$$\frac{d\mathbf{u}}{dt} = \mathbf{f}(\mathbf{u}) - BK\mathbf{u} \tag{3.2}$$

where $\mathbf{f}(\mathbf{u})$ is the non-linear vector arising from the finite difference discretisation of the right hand side of (3.1), $\mathbf{u}(t) = \begin{bmatrix} u_1(t) & u_2(t) & \dots & u_N(t) \end{bmatrix}^T$ is the state vector of length N, and K is the $N \times N$ feedback control matrix. For shadowing control, we seek the optimal values of K that minimise the following objective

$$\overline{J^{(\infty)}} = \lim_{T\to\infty} \frac{1}{TL} \int_0^T \left(\mathbf{u}^T \delta x \mathbf{u} + \frac{\alpha}{2} (K\mathbf{u})^T \delta x (K\mathbf{u}) \right) dt \tag{3.3}$$

subject to the non-linear constraint (3.2). The objective and constraint have the same form as (2.11). The first term on the right hand side of (3.3) represents the space-time average kinetic energy of the system, while the second term represents the cost of the control effort, which is regulated by the parameter α. We use N equally spaced actuators (equivalent to setting $B = I$ in equation 3.2), and we search for the optimum values of K that drive $\mathbf{u}(t)$ to zero, i.e. that bring the system to rest.



(a) Time average of $u(x,t)$

(b) Temporal RMS of $u(x,t)$

Figure 2: Time average and RMS of $u(x,t)$. $\langle\tilde{u}\rangle$ and $\tilde{u}_{RMS}$ were obtained for trajectories with length $T=2000$ and averaged over 150 random initial conditions in $[0,1]$.

In the following, we compare this matrix K with the one obtained using the LQR algorithm. For the latter, the linear version of the KSE (3.1), i.e.

$$\frac{\partial u}{\partial t} = -\frac{\partial^2 u}{\partial x^2} - \frac{\partial^4 u}{\partial x^4} \quad x \in [0, L] \tag{3.4}$$

with the same boundary conditions is discretised to obtain the linearised matrix A_l, which is required to compute the LQR feedback matrix, (2.10). We then apply both matrices to the full non-linear discretised KSE (3.2).

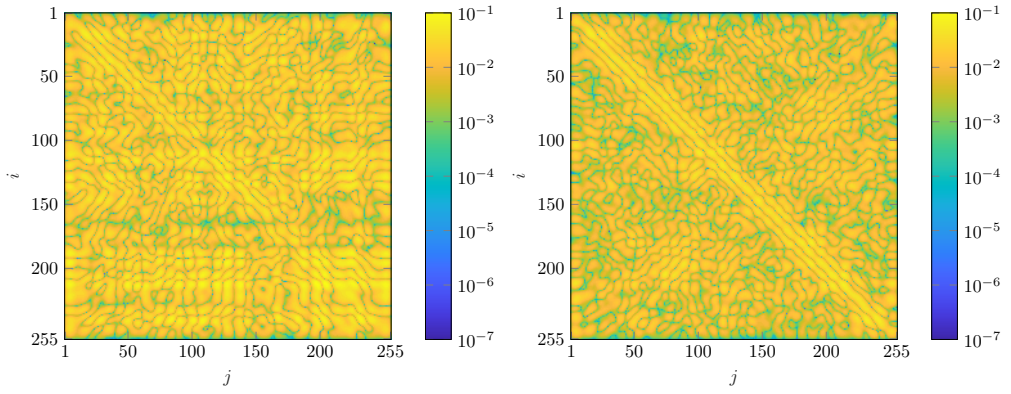
(a) Comparison of PMSS Control and LQR control kernels

Since we have selected $B=I$, the actuation $\mathbf{s}(t) = -K\mathbf{u}(t)$ can be written explicitly as $s_i(t) = -\sum_{j=1}^{N} K_{i,j}u_j(t)$, where the index i corresponds to the control location, $x_c = i \times \delta x$. Written in this form, the physical meaning of $K_{i,j}$ becomes clear; it represents the weight of the j-th velocity to the actuation in the i-th point. Summing all j contributions, results in $s_i(t)$. For more general cases, the input matrix B provides an appropriate spatial weighting and the control signal is given by $B\mathbf{s}$, see equation (2.2).

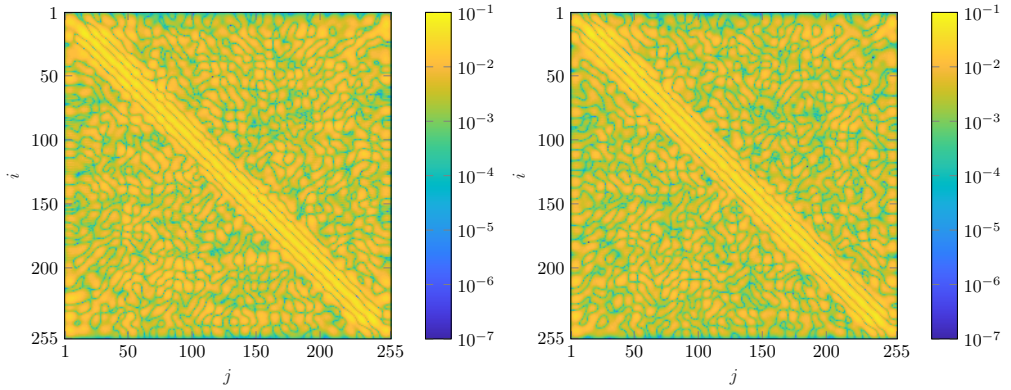
The total derivative of $\overline{J^{(T)}}$ with respect to element $K_{i,j}$ is found using (2.27). For the objective function (3.3) considered here, (2.27) translates into

$$\frac{d\overline{J^{(T)}}}{dK_{i,j}} = -\left(\sum_{p=1}^{P}\int_{t_{p-1}}^{t_p}\left(u_j(t)\hat{w}_i(t)\right)\,dt\right) + \frac{\alpha\delta x}{TL}\int_0^T\left(u_j(t)\sum_{n=1}^{N}u_n(t)K_{i,n}\right)\,dt \tag{3.5}$$

Figure (3) shows the absolute values of the elements of the sensitivity matrix $\left|\frac{d\overline{J^{(T)}}}{dK_{i,j}}\right|$ in log-scale, for different trajectory lengths T. For $T=50$ (panel a), the matrix does not seem to have a clear structure. For $T=200$ (panel b), the matrix is starting to acquire a diagonally dominant structure. However a significant number of elements in the top right and bottom left quadrants are of the same order of magnitude $O(10^{-1})$ as the central diagonals (shown in yellow). Time horizons $T=50$ and $T=200$ are clearly not long enough for sensitivity convergence. However, increasing the trajectory length to $T=500$ (panel c) and $T=800$ (panel d), the sensitivities are starting to converge and become independent of T. It is now clear that the matrix $\left|\frac{d\overline{J^{(T)}}}{dK}\right|$ has a diagonally dominant structure. Note that convergence is first attained around the main diagonal and slowly propagates further away.



(a) $T = 50$ (b) $T = 200$



(c) $T = 500$ (d) $T = 800$

Figure 3: Colour maps of the absolute values of sensitivities $\left|\frac{d\overline{J^{(T)}}}{dK}\right|$ in log-scale for different time-averaging lengths T. They are obtained from the attractor of the uncontrolled system (i.e. the first iteration with $K^{(0)} = 0$).

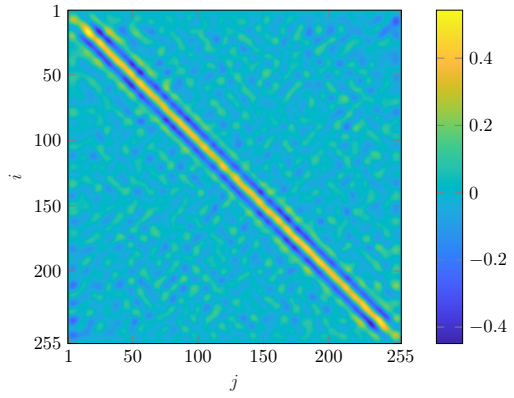


Figure 4: Colour map of matrix K obtained using PMSS control with $T = 800$.

Using the computed sensitivities for $T = 800$ and a step size $a^{(0)} = 10$, we compute the elements $K^{(1)}_{i,j}$ from (2.14) and plot the colour map in linear scale in Figure (4). Large positive and negative values are found around the main diagonal, while further away i.e. when $j \gg i$, $K_{i,j}$ decays to 0. This has a clear physical meaning; the value of actuation at point i is determined mainly by the nearby neighbours, while the contribution of points located further away becomes progressively smaller and smaller. The control kernel therefore has compact support. This has been demonstrated for the kernel obtained when LQR is applied to the linearised Navier-Stokes equations in a channel flow (see Figure (6) of [38]). The present analysis shows that the same property holds if we compute K directly from the nonlinear attractor, at least for the case examined. More research is needed however to determine if controllers with compact support can be computed for general non-linear systems using the current algorithm.

Averaging along the m-th diagonal of K (the main diagonal corresponds to $m = 0$), we obtain $\overline{diagK}(\xi)$ which depends only on $\xi = m\delta x = x - x_c$. It is clear that $\overline{diagK}(\xi)$ represents the average value of weights against the distance ξ from the actuation point. We plot the distribution of $\overline{diagK}(\xi)$ for two grid spacings δx in Figure (5a) with solid lines. We superimpose also the weights from LQR obtained with control weight $c = 1$ (refer to equation 2.10) with dashed lines. We notice that the weights obtained from LQR and PMSS have very similar distributions. As expected, they are both localised around $\xi = 0$ and decay to 0 further away. Also they both depend on the discretisation.

In order to further analyse the results, we compute the control kernel, which is independent of δx. To this end, we write the actuation at location x_c as the convolution integral

$$s(x_c) = -\int_0^L \mathscr{K}\,(x - x_c; x_c)\, u(x)\, dx \tag{3.6}$$

where $\mathscr{K}\,(x - x_c; x_c)$ is the convolution (or control) kernel at x_c. This can be computed from the matrix elements $\mathscr{K}\,(x - x_c; x_c) = \mathscr{K}\,(\xi; x_c) = \frac{K_{i,j}}{\delta x}$, where $x_c = i \times \delta x$ and $x = j \times \delta x$. In Figure (5b) we plot both diagonally-averaged kernels and, in order to facilitate the comparison, we zoom in the region $\xi \in [-12, 12]$. The LQR kernels (dashed lines) collapse perfectly for the two discretisations, as they should. The PMSS kernels however (solid lines), although close, they do not collapse to a single curve. Perhaps we should not expect them to collapse, as they are found directly from the non-linear attractor under the assumption of the feedback control law (2.3). The shapes produced by the two control methods are similar, but the LQR kernel decays to 0 faster than the PMSS kernel. The latter has more pronounced peaks and oscillates around 0.

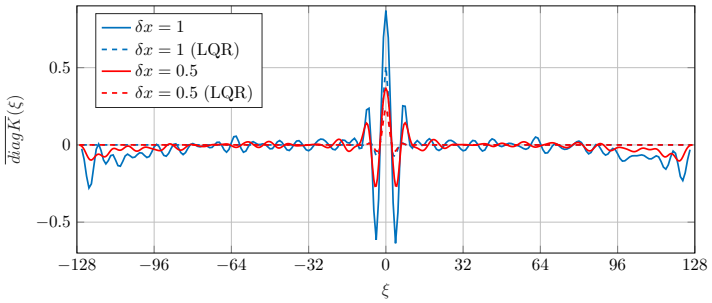
To eliminate the effect of the step size $a^{(0)}$ and the control cost parameter c that affect the absolute values of $\overline{\mathscr{K}}(\xi)$, in Figure (5c) we plot the kernel distribution normalised with the value at $\xi = 0$. This normalisation reveals a very interesting feature; the two kernels are almost identical in the region $-2.5 \le \xi \le 2.5$, but deviate elsewhere.

What determines the shape of the kernels and why do they have this distribution? In order to obtain more insight, we consider the two-point spatial correlation function at a given location x_c

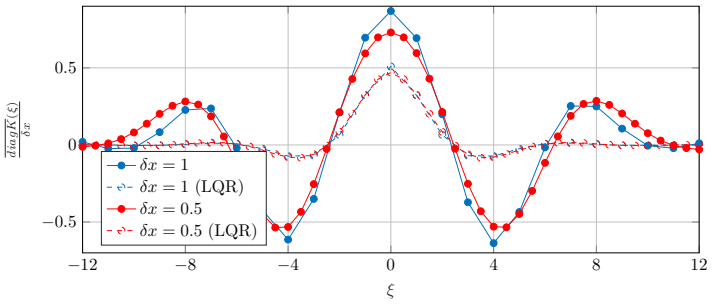
$$\rho(\xi; x_c) = \frac{\langle u'(x_c, t) u'(x_c + \xi, t)\rangle}{\langle u'(x_c, t)^2\rangle} \tag{3.7}$$

where $u' = u - \langle u\rangle$ is the fluctuation about the time-average $\langle u\rangle$. A small correlation $\rho(\xi; x_c) \approx 0$ indicates that a perturbation at $x_c + \xi$ (for example due to actuation), would not be 'felt' at x_c. It is therefore expected that this function will be related to the control kernel.

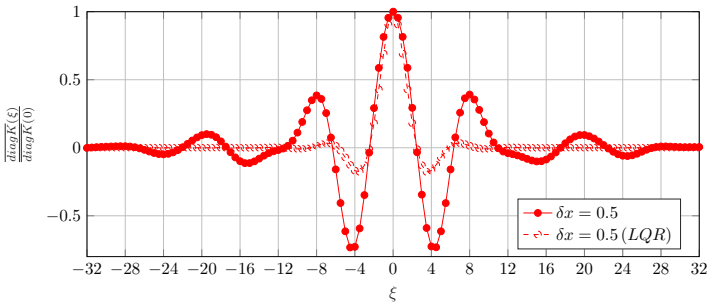
We plot $\rho(\xi; x_c)$ against ξ at different locations x_c along the domain in Figure (6a). The correlations collapse very well for $\xi \in [-4, 4]$, but start to deviate as ξ becomes larger. Note also the slight loss of symmetry around $\xi = 0$ for points x_c close to the boundaries of the domain. We then average $\rho(\xi; x_c)$ over x_c in the region $40 < x_c < 90$ and plot the distribution of the average correlation $\overline{\rho(\xi)}$ in Figure (6b). Symmetry around $\xi = 0$ has now been restored. Distinct positive and negative peaks can be identified at $\xi \approx \pm 4, \pm 8.5, \pm 13.5$ etc. Moreover $\overline{\rho(\xi)}$ decays to zero for $\xi < -20$ and $\xi > 20$.



(a) Full domain



(b) Mean convolution kernel in the region $-12 \leq \xi \leq 12$

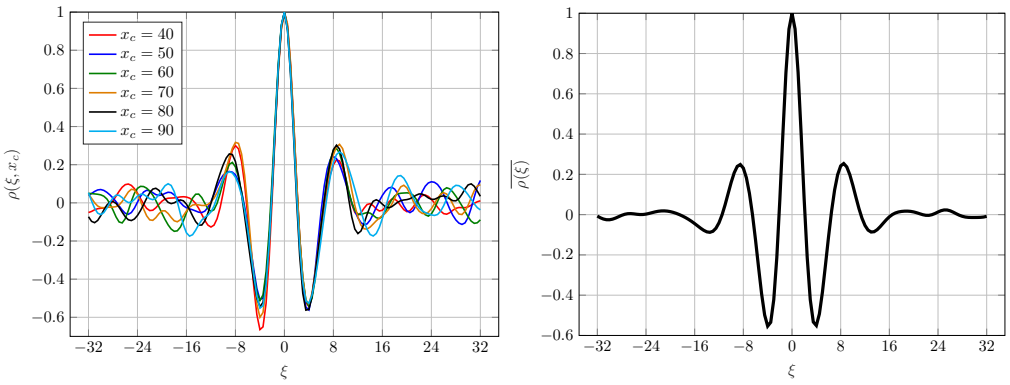


(c) Mean convolution kernel normalised by $\frac{\overline{diagK}(0)}{\delta x}$

Figure 5: Distribution of feedback matrix weights (Panel a) and kernels (Panels b,c) obtained by averaging along the diagonals of K and plotting against $\xi = x - x_c$.

We can physically explain these results by reference to Figure (1). The peaks at $\xi = \pm 4$ indicate the average distance between positive and negative streaks that are located next to each other. The fluctuations around the average $\langle u \rangle$ have opposite signs and therefore $\overline{\rho(\pm 4)} < 0$. The lower peaks at $\xi = \pm 8.5$ indicate weaker correlation between two positive or two negative streaks. The even lower negative peak at $\xi = \pm 13.5$ can be explained similarly. The results indicate that the correlation is strong over approximately $\xi = 8.5$, i.e. over the average distance between streaks of the same sign.

In Figure (7) we plot together the average $\overline{\rho(\xi)}$ and the normalised LQR and PMSS control kernels. Plotted in this way, all three distributions have remarkable similarities, but also some differences. The two control kernels and the correlation overlap in $-2.5 < \xi < 2.5$. At $\xi = \pm 4$, the troughs are more clearly pronounced for the PMSS kernel and are closer to $\overline{\rho(\pm 4)}$. On



(a) Correlations $\rho(\xi; x_c)$ against ξ for different x_c

(b) Averaged correlation in the region $x_c \in [40, 90]$

Figure 6: Two-point spatial correlations.

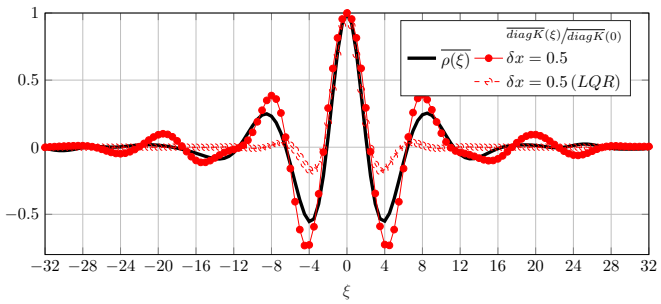


Figure 7: Mean two-point correlation $\overline{\rho(\xi)}$ superimposed on $\frac{\overline{diagK}(\xi)}{\overline{diagK}(0)}$

the other hand, the LQR kernel decays very quickly outside the ξ region $[-4, 4]$. The narrow support of the LQR kernel indicates that the actuation acts to annihilate the positive/negative streaky combinations. On the other hand, the PMSS kernel has wider support and opposes larger positive/negative/positive streaky combinations.

(b) Response of the controlled system

Next we turn our attention to the response of the system (3.2) to PMSS and LQR actuations. We consider the instantaneous, spatially-averaged kinetic energy,

$$J(t) = \frac{1}{L}\left(\mathbf{u}(K,t)^T \delta x \mathbf{u}(K,t)\right) \tag{3.8}$$

and explore how fast $J(t)$ is reduced. For PMSS, we apply control with different kernel sizes, i.e. we select a ξ_{max} and compute the actuation using only the elements of K in the region $-\xi_{max} \le \xi \le \xi_{max}$ around each control point x_c. We set the step size $a^{(0)} = 6$ on step 5 of the control algorithm. To ensure a fair comparison, we choose (by trial and error) a value $c = 1.1$ in equation (2.10) so that the average of the main diagonal, $\overline{diag(K)}(0)$, is the same for PMSS and LQR.

The energy $J(t)$ is plotted against time in Figure (8). For $a^{(0)} = 6$, all controllers bring the system to rest and so no additional iterations are required. It is remarkable that even a single PMSS

control iteration, which relies only on information of the uncontrolled system, is so effective. When $\xi_{max} = 2$, the normalised LQR and PMSS matrix kernels are very similar (as shown in Figure 7), and hence $J(t)$ drops at the same rate for both controllers. A contour plot of the controlled solution $u(t)$ is shown in Figure (9) and shows how effective the actuation is. It can be clearly seen that the streaks found in the uncontrolled case (Figure 1) are rapidly annihilated upon application of the control and $|u(x,t)|$ is brought to 0 within $5-10$ time units.

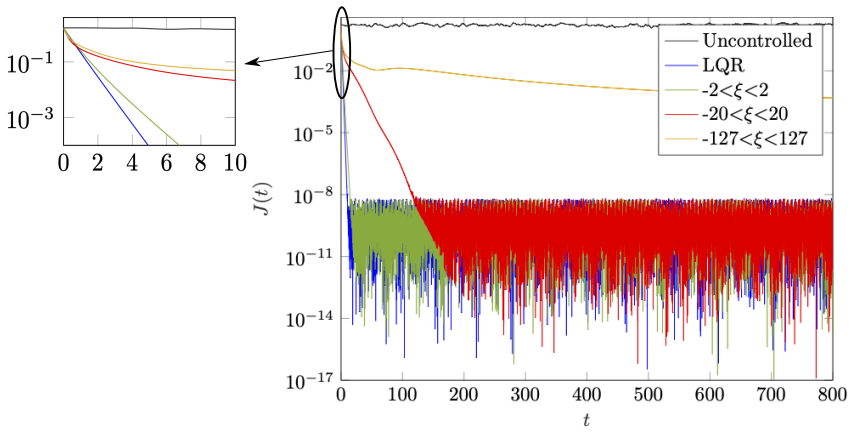


Figure 8: Instantaneous kinetic energy $J(t)$ of the actuated system using PMSS and LQR. The uncontrolled case is shown in black colour and the LQR in blue. For the PMSS control matrix K, we use only the elements that fall inside the indicated range of ξ. Decreasing ξ leads to faster stabilisation.

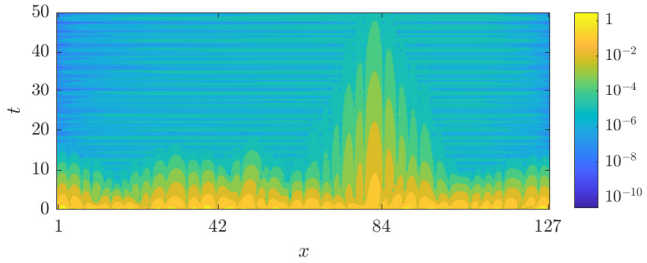
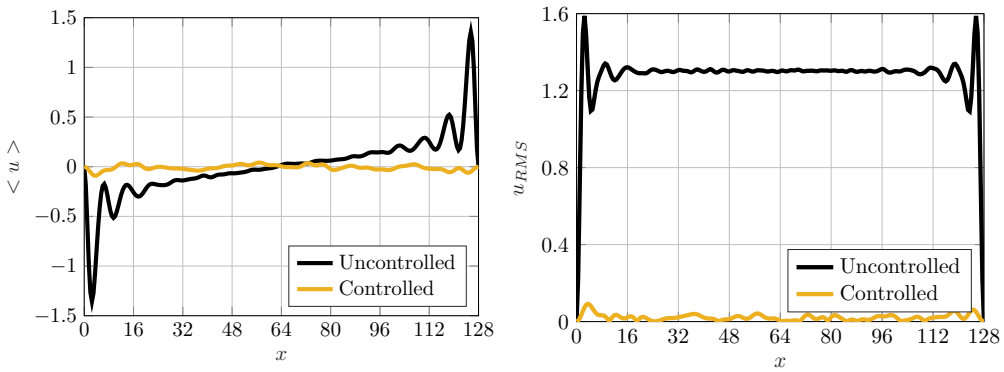


Figure 9: Absolute values of the controlled solution $|u(x,t)|$ in log-scale (with $\xi_{max} = 2$).

Increasing ξ_{max} to 20 reduces the rate at which the system is brought to rest. Using the full matrix, i.e. setting $\xi_{max} = 127$, still reduces the energy by 4 orders of magnitude (from $J(0) = 1.68$ to $J(T) = 4.8 \times 10^{-4}$). However, the response of the system is slow when $J(t)$ falls below 10^{-2}, i.e. this is a long-term effect.

Indeed, as can be seen from Figure (8), and especially from the inset that zooms-in to small values of t, the rate of descent is initially the same for PMSS and LQR. Most importantly, for the PMSS controller this holds for all values of ξ_{max}, even the largest. The curves start to deviate approximately for $t > 1$. In Figure (10) we plot the spatial distribution of the time-average and rms of $u(x,t)$ with and without control (the results are with $\xi_{max} = 127$). Note the effectiveness in suppressing both variables and bringing them close to 0 across the whole domain.

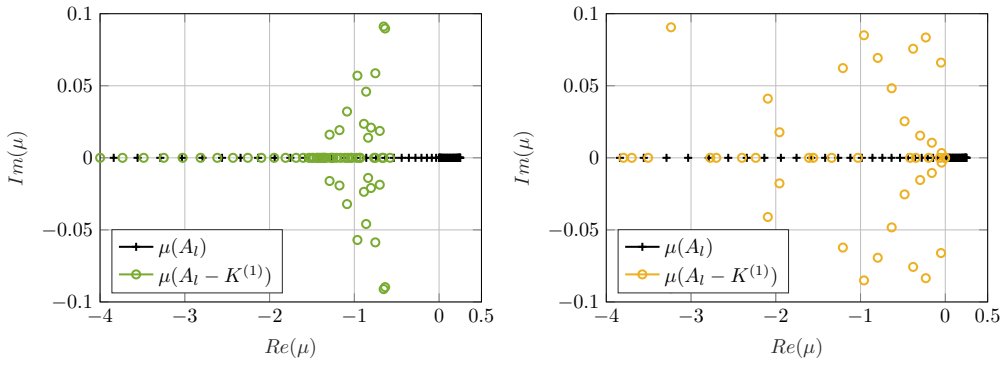


(a) Time average of $u(x,t)$

(b) RMS of $u(x,t)$

Figure 10: Time average and RMS of $u(x,t)$. The controller $K^{(1)}$ uses the full matrix, i.e. with $-127 \leq \xi \leq 127$. The statistics are computed by time-averaging between $t = 500$ and $t = 800 (= T)$.

In order to shed more light onto the behaviour shown in Figure (8), we consider the matrix $A_l - K^{(1)}$, where A_l is the discrete form of the right hand side of the linearised equation (3.4) and compute its eigenvalues. Figure (11) shows the eigenvalues, μ, for two values of $\xi_{max} = 2, 127$. Note that for the uncontrolled case, all $\mu(A_l)$ are real because the matrix A_l is symmetric (contains only second and fourth order derivatives that are discretised with central differences). However, some $\mu(A_l - K^{(1)})$ have a small imaginary part due to lack of strict symmetry of $K^{(1)}$. This is likely due to the finite length of the trajectory T and the slow convergence of the off-diagonal elements. Some eigenvalues of A_l have positive real parts, indicating linear instability. When the controller is introduced, all eigenvalues move to the left hand plane, i.e. become stable. However for $\xi_{max} = 127$, some eigenvalues are close to the imaginary axis, explaining the slower rate of descent for $J(t)$.



(a) $K^{(1)}$: $-2 \leq \xi \leq 2$

(b) $K^{(1)}$: $-127 \leq \xi \leq 127$

Figure 11: Eigenvalues of the controlled and uncontrolled matrices $A_l - K^{(1)}$ and A_l respectively, plotted in the complex-plane ($K^{(1)} = -a^{(0)} \overline{\frac{dJ(T)}{dK}}^{(0)}$ for $T = 800$ and $a^{(0)} = 6$). Only values with $Re(\mu) > -4$ are shown.

4. PMSS control algorithm performance

We now look at the performance of the PMSS control algorithm. We set as inputs $T = 50$, $K^{(0)} = 0$, and $\epsilon = 1 \times 10^{-2}$, while restricting K to the diagonals in the region $-2 \leq \xi \leq 2$. A number of performance measures are given in Table (1). It is clear that a single iteration is sufficient to drive the kinetic energy evaluated at $t = T$, $J(T)$, to zero. Since one iteration renders the controlled system linearly stable, i.e. $Re\big(\mu_{max}\big(A_l - K^{(1)}\big)\big) < 0$, a second iteration is unnecessary.

Iteration #	$\overline{J^{(T)}}$	$J(T)$	$\|\frac{d\overline{J^{(T)}}}{dK}^{(i)}\|_2$	$a^{(i)}$	# GMRES Iterations
$i = 0$	1.62	1.37	3.7×10^{-1}	10	34
$i = 1$	1.4×10^{-2}	$\approx 1 \times 10^{-10}$	1×10^{-2}	10	2
$i = 2$	1.4×10^{-2}	$\approx 1 \times 10^{-10}$	-	-	-

Table 1: Some key PMSS control algorithm performance measures. The algorithm inputs used are $T = 50$, $K^{(0)} = 0$, and $\epsilon = 1 \times 10^{-2}$. K is restricted to $-2 \leq \xi \leq 2$.

The time-average over $0 \to T$, $\overline{J^{(T)}}$, is reduced by 2 orders of magnitude between the $0 - th$ iteration (uncontrolled flow) and the controlled flow after one iteration. The actual value of $\overline{J^{(T)}}$ depends on T and accounts for the transient shown in Figure (8). The variation of $\overline{J^{(T)}}$ against T (when the algorithm is run separately for each T) is shown in Figure (12). $\overline{J^{(T)}}$ drops with rate $\sim T^{-1}$, because the transient period occupies a smaller and smaller fraction of T as $T \to \infty$. The 2-norm of the sensitivity matrix $\|\frac{d\overline{J^{(T)}}}{dK}^{(i)}\|_2 = \sigma_{max}\big(\frac{d\overline{J^{(T)}}}{dK}^{(i)}\big)$, where σ_{max} denotes the maximum singular value, drops by a factor of three from the $0 - th$ to the 1^{st} iteration. Again, this depends on T.

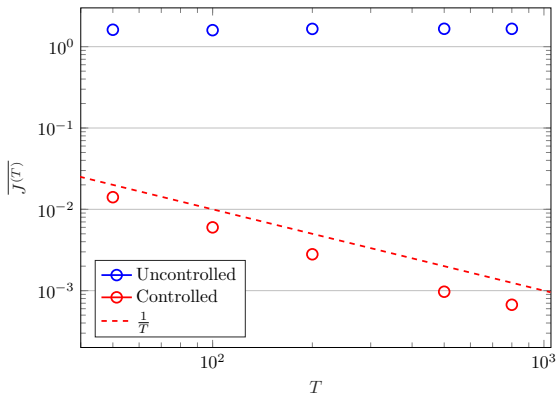


Figure 12: Space-time averaged objective $\overline{J^{(T)}}$ obtained by running the PMSS algorithm for different T; it drops with rate $\sim T^{-1}$.

Table (2) shows the PMSS solver parameters (step 3 of the control algorithm). Initially the uncontrolled trajectory ($K^{(0)} = 0$) has 15 positive Lyapunov exponents ($N_{+LE} = 15$). We therefore apply a preconditioner with $l = 15$ on the first iteration (to annihilate these fastest growing modes). The condition number κ is reduced by 4 orders of magnitude and the convergence of GMRES is achieved in 34 iterations. On the second iteration all unstable modes have been annihilated ($N_{+LE} = 0$), and therefore no preconditioning is necessary (the condition number $\kappa(S) = 5$ is already very small).

Iteration #	N_{+LE}	ΔT	l	q	γ	$\kappa(S)$	$\kappa(H)$	# GMRES Iterations
$i=0$	15	5	15	1	0.1	1.3×10^5	24	34
$i=1$	0	5	-	-	0.1	5	-	2

Table 2: PMSS solver parameters. N_{+LE} refers to the number of positive exponents of the trajectory on step 3 of the control algorithm. $\kappa(S)$ and $\kappa(H)=\kappa\left(\gamma I+\mathbf{M}_{\mathbf{BD}}^{(q)}{}_{(l)}S\right)$ are the condition numbers of the unconditioned and preconditioned MSS matrices, respectively.

For the PMSS control algorithm to be applicable to large systems, it must be able to compute accurate sensitivities as efficiently as possible. Step 3 of the PMSS solver i.e. the solution of the linear system (2.28), dominates the computational cost, making preconditioning necessary for scalability. In Figure (13) we plot the convergence rate quantified in terms of GMRES residuals ($\|r_{(m)}\|_2=\|\underline{q}-H\underline{\hat{\mathbf{w}}}_{(m)}\|_2$), against the GMRES iteration number m, where $\underline{q}$ and H are, respectively, the right hand side vector and the matrix of (2.28) for different trajectory lengths T. It is clear that the dependence of the convergence rate on T is very weak. The residuals drop 5 orders of magnitude in between 25-37 iterations, even though T varies by more than an order of magnitude, from $T=50$ to $T=800$, increasing the number of unknowns $\underline{\hat{\mathbf{w}}}$ by a factor of 16.

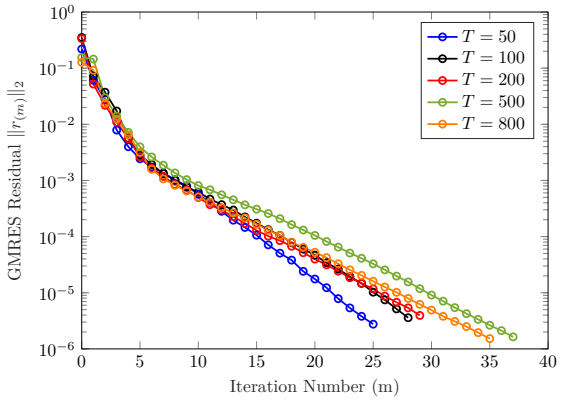


Figure 13: GMRES Residuals $\|r_{(m)}\|_2$ for Step 3 of the PMSS solver. The preconditioner parameters used are $\Delta T=10$, $l=15$, $q=1$ and $\gamma=0.1$. The residual $\|r_{(m)}\|_2$ drops by approximately five orders of magnitude by the final iteration for all T values.

5. Conclusions

We have proposed an algorithm that couples shadowing adjoint sensitivity analysis with gradient descent to compute the feedback control matrix K for a chaotic system. The sensitivities were used as search directions to find the matrix elements that minimise an objective function, subject to the full non-linear system constraints. Most importantly, due to the adopted adjoint formulation, the computational cost is independent of the number of the matrix elements. We applied this approach to the Kuramoto Sivashinsky equation (KSE), and managed to stabilise it around the rest position, $\mathbf{u}(t)=0$. It was shown that for suitably chosen parameters, a single iteration of the algorithm was sufficient to compute a stabilising feedback matrix K.

We compared the control kernels obtained with this algorithm and the standard LQR approach and noted similarities and differences. Both kernels had compact support and similar shape, which was related to the streaky structure of the solution of the uncontrolled KSE. They were almost identical for short separations from the actuation point, but the LQR kernel decayed faster

to 0. This difference is most likely due to the nonlinear terms that are ignored in LQR. All kernels computed with the PMSS algorithm were stabilising.

From a computational point of view, the cost of LQR scales with $O(N^3)$, which poses severe restrictions for application to large scale systems. On the other hand, the PMSS algorithm uses only time steppers and a preconditioner to make the convergence almost independent of N and T. The cost of the preconditioner depends however on the number of positive Lyapunov exponents. While the LQR has shown faster stabilisation of the instantaneous energy for the case examined (see Figure 8), the performance of PMSS was almost as good for restricted kernels.

We close this section with some thoughts on possible extensions of this work. In practical applications, only noisy measurements at some locations, say $\mathbf{y}_j(t)$, are available. This is known as output feedback control; in this case the actuation must be expressed as $s_i(t) = G(\mathbf{y}_j)$. Assuming G to be a linear function of present and past values of the measurements (to account for memory effects), we can write $s_i(t) = K_0\mathbf{y}_j(t) + K_1\mathbf{y}_j(t-\delta t) + ... + K_n\mathbf{y}_j(t-n\delta t)$, where the matrices $K_0, K_1, .., K_n$ can be obtained with the same method presented in this paper (instead of neural networks as for example in [31]). The value of n is problem dependent, and be estimated from the time autocorrelation of the uncontrolled solution.

The proposed approach is also scalable to 2D and 3D systems and requires only time-steppers of the full nonlinear as well as the linearised forward and adjoint equations as mentioned above. It would be instructive to apply the method to other systems, for example suppression of vortex shedding around a cylinder at transitional Reynolds numbers, the recirculation zone in the suction suction of an airfoil, etc.

Data Accessibility. The MATLAB code used to generate the results in this paper can be accessed from https://drive.google.com/file/d/1lVmx2wqZWi4PTktCef8oN0ERvWvy1oaA/view?usp=sharing

Funding. The first author has received a PhD scholarship from Al-Alfi Foundation.

Acknowledgements. The authors would like to thank the anonymous reviewers for their useful comments.

References

1 Gunzburger MD. 2002 *Perspectives in Flow Control and Optimization*. USA: Society for Industrial and Applied Mathematics.
2 Xiao D, Papadakis G. 2017 Nonlinear optimal control of bypass transition in a boundary layer flow. *Physics of Fluids* **29**, 054103.
3 Luchini P, Bottaro A. 2014 Adjoint Equations in Stability Analysis. *Annual Review of Fluid Mechanics* **46**, 493–517.
4 Bryson A, Ho YC, Siouris G. 1975 *Applied Optimal Control*. Hemisphere Pub. Corp.
5 Green M, Limebeer D. 1995 *Linear Robust Control*. Prentice Hall, Englewood Cliffs.
6 McKernan J, Papadakis G, Whidborne JF. 2009 Linear and non-linear simulations of feedback control in plane Poiseuille flow. *Int. J. Num. Mech. Fluids* **59**, 907–925.
7 Barbagallo A, Sipp D, Schmid P. 2009 Closed-loop control of an open cavity flow using reduced-order models. *J. Fluid Mech.* **641**, 1–50.
8 Bagheri S, Brandt L, Henningson D. 2009 Input-output analysis, model reduction and control of the flat plate boundary layer. *J. Fluid Mech.* **620**, 263–298.
9 Semerano O, Bagheri S, Brandt L, Henningson DS. 2011 Feedback control of three-dimensional optimal disturbances using reduced-order models. *J. Fluid Mech.* **677**, 63–102.
10 Sharma AS, Morrison JF, McKeon BJ, Limebeer DJN, Koberg WH, Sherwin SJ. 2011 Relaminarisation of Re_τ=100 channel flow with globally stabilising linear feedback control. *Physics of Fluids* **23**, 125105.
11 Heins PH, Jones BL, Sharma AS. 2016 Passivity-based output-feedback control of turbulent channel flow. *Automatica* **69**, 348–355.
12 Pralits J, Luchini P. 2010 Riccati-less optimal control of bluff-body wakes. In Schlatter P, Henningson D, editors, *Proc. 7th IUTAM Symposium on Laminar-Turbulent Transition* Stockholm, Sweden. Springer.
13 Bewley T, Luchini P, Pralits J. 2016 Methods for solution of large optimal control problems that bypass open-loop model reduction. *Meccanica* **51**, 2997–3014.

14 Semerano O, Pralits J, Rowley C, Heningson D. 2013 Riccati-less approach for optimal control and estimation: an application to two dimensional boundary layers. *J. Fluid Mech.* **731**, 394–417.
15 Semeraro O, Pralits JO. 2018 Full-order optimal compensators for flow control: the multiple inputs case. *Theoretical and Computational Fluid Dynamics* **32**, 285–305.
16 Mårtensson K, Rantzer A. 2012 A scalable method for continuous-time distributed control synthesis. In *2012 American Control Conference (ACC)* pp. 6308–6313.
17 Mårtensson K. 2012 *Gradient Methods for Large-Scale and Distributed Linear Quadratic Control*. PhD thesis Department of Automatic Control, Lund Institute of Technology, Lund University.
18 Wang Q, Hu R, Blonigan P. 2014 Least Squares Shadowing sensitivity analysis of chaotic limit cycle oscillations. *Journal of Computational Physics* **267**, 210–224.
19 Lasagna D. 2017 Sensitivity Analysis of Chaotic Systems using Unstable Periodic Orbits. *SIAM J. Applied Dynamical Systems* **17**, 547–580.
20 Blonigan PJ, Wang Q. 2018 Multiple shooting shadowing for sensitivity analysis of chaotic dynamical systems. *Journal of Computational Physics* **354**, 447–475.
21 Ni A, Wang Q. 2017 Sensitivity analysis on chaotic dynamical systems by Non-Intrusive Least Squares Shadowing (NILSS). *Journal of Computational Physics* **347**, 56–77.
22 Shawki K, Papadakis G. 2019 A preconditioned multiple shooting shadowing algorithm for the sensitivity analysis of chaotic systems. *Journal of Computational Physics* p. 108861.
23 Blonigan PJ. 2017 Adjoint sensitivity analysis of chaotic dynamical systems with non-intrusive least squares shadowing. *Journal of Computational Physics* **348**, 803–826.
24 Blonigan PJ, Wang Q. 2014 Least squares shadowing sensitivity analysis of a modified Kuramoto-Sivashinsky equation. *Chaos, Solitons and Fractals* **64**, 16–25.
25 Günther S, Gauger NR, Wang Q. 2017 A framework for simultaneous aerodynamic design optimization in the presence of chaos. *Journal of Computational Physics* **328**, 387–398.
26 Cagliari LV, Mishra S, Hicken JE. 2019 Plant and controller optimization in the context of chaotic dynamical systems. *AIAA Scitech 2019 Forum, 0167* pp. 1–21.
27 Armaou A, Christofides PD. 2000 Feedback control of the Kuramoto–Sivashinsky equation. *Physica D: Nonlinear Phenomena* **137**, 49 – 61.
28 Christofides PD, Armaou A. 2000 Global stabilization of the Kuramoto-Sivashinsky equation via distributed output feedback control. *Systems and Control Letters* **39**, 283–294.
29 Gomes SN, Pradas M, Kalliadasis S, Papageorgiou DT, Pavliotis Ga. 2015 Controlling spatiotemporal chaos in active dissipative-dispersive nonlinear systems. *Physical Review E* **92**, 022912.
30 Gomes SN, Papageorgiou DT, Pavliotis GA. 2017 Stabilizing non-Trivial solutions of the generalized Kuramoto-Sivashinsky equation using feedback and optimal control. *IMA Journal of Applied Mathematics (Institute of Mathematics and Its Applications)* **82**, 158–194.
31 Bucci MA, Semeraro O, Allauzen A, Wisniewski G, Cordier L, Mathelin L. 2019 Control of chaotic systems by deep reinforcement learning. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* **475**, 20190351.
32 Astrom KJ. 2006 *Introduction to Stochastic Control Theory*. Dover, New York.
33 Brent RP. 1973 *Algorithms for Minimization without Derivatives*. Prentice-Hall.
34 Peres PLD, Geromel JC. 1994 An alternate numerical solution to the linear quadratic problem. *IEEE Transactions on Automatic Control* **39**, 198–202.
35 Saad Y. 2003 *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics 2 edition.
36 Baglama J, Reichel L. 2005 Augmented Implicitly Restarted Lanczos Bidiagonalization Methods. *SIAM Journal on Scientific Computing* **27**, 19–42.
37 Holmes P, Lumley JL, Berkooz G, Rowley CW. 2012 *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*. Cambridge: Cambridge University Press 2nd edition.
38 Hogberg M, Bewley TR, Henningson DS. 2003 Linear feedback control and estimation of transition in plane channel flow. *J. Fluid Mech.* **481**, 149–175.