Data-driven modelling and predictive control of fluid flows

by

Luigi Marra

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in

Aerospace Engineering

Universidad Carlos III de Madrid

Advisors:
Stefano Discetti
Andrea Meilán-Vila

Tutor:
Stefano Discetti

Leganés, March 2026

To Paola, and to my family

Acknowledgements

And here I am, after several years, facing perhaps the most delicate part of this journey. A section where there are no supervisors, no peer reviews, no Google Scholar to draw references from. A section made only of memories, emotions, and moments that will remain unforgettable. These lines do not belong solely to this thesis, which marks the end of a journey, but to a chapter of the book of my life of which I am immensely proud. Not so much for the achievements reached, but for the people who have accompanied me along the way and whom I hope will continue to do so. The time spent with you has taken on a different dimension, almost scientifically inexplicable: these years have flown, perhaps because you expanded my life.

First, I want to thank the people who made possible this adventure, which began several years ago. Thank you to my supervisors, Stefano and Andrea, for your daily, academic, and personal support. Without your constant presence, neither this PhD nor I would have been the same. I have always admired your enthusiasm, your perseverance, and the energy with which you have always sought to do more, and to do it better. Thank you for recognising and valuing my potential, but above all, for helping me understand my limits and overcome them. Thank you for your friendship, for your trust, and for teaching me to see every goal as something attainable. I could not have wished for better mentors, neither academically nor personally.

In the end, we are always surrounded by people to learn from, to take as an example. Among these, I want to thank Andrea I., Marco, Rodrigo, and Andrea C. for being a constant point of scientific and personal reference for me. Thank you for the moments spent together, inside and outside the university, and for always finding time for advice or a word of encouragement. The best conversations were born while sitting at a bar or riding a bike. I cannot fail to mention, as well, Onofrio and Lionel. Thank you for the depth of your teachings, for providing a turning point to my PhD journey, and for teaching me to see things from a different perspective. Thank you for your enthusiasm and passion for science, but also for sharing interests that went far beyond work. I hope to return to our cultural coffee breaks one day.

Thank you to all the people I have been lucky enough to meet at the University.

A special thanks goes to the entire EAP Lab group: past and present, from PhD students to professors, all the way to program managers. After all, a family is also made up of the people we choose to have by our side. Thank you for every shared moment, from scientific discussions to conferences to social events of every kind. A special thanks to Olympia for guiding me with patience through the countless bureaucratic adventures.

Thanks to Mr Chen, a man of few words, yet always ready to amaze everyone with his brilliant ideas.

According to some studies, playful insults are a fundamental ingredient for a sincere and lasting friendship. If that is the case, one of the people with whom I have cultivated the deepest bond is, without a doubt, Iacopo, or more recently, "o' posdocc". You were the first person I met, even before moving to Madrid, and if I have made it this far, it is largely thanks to your teachings as an older brother. Thank you for the understanding, the support, and (I admit, at times) the patience, but above all, for your constant presence.

Alongside Iacopo, I cannot fail to mention Luca S., or "Scheli," as he has always been called. Companion of "grosselli" and "sushini", a true point of reference in this adventure. If I were asked to name my most sincere friends, your name would surely be at the very top of the list.

Thanks to Gabriele, a truly genuine friend, always ready to offer the right advice at the right time, companion in unforgettable exploratory adventures, from the summer Pantano to Manzanares el Real (where perhaps visiting the castle would have been worth it!) and to skiing days.

Thanks to Enrico A., friend in name, but above all, in fact. I will never forget our scientific discussions with the "Azkaban crew".

Thanks to all the members of 'Oficina 1.2.A.06a', from the beginning with Patri and Alvaro to Alicia, Qi, Pablo, and Tani. I will never forget the Telepizza lunches, the long chats, and the intense workdays, made lighter by a most improbable soundtrack between villancicos, Pavarotti, and rock. Thank you, Alicia, for constantly accompanying me on this journey, even in the most tortuous moments, and for patiently tolerating every madness.

Thanks to Matteo R. for the endless discussions on science and general culture in Cercanías. I have a profound admiration for you! I also hope that one day I will be able to block one of your volleyball spikes.

An immense thanks goes to the guys from 'Plaza del Campillo del Mundo Nuevo 1', Luca, Donato (and Mich). I could not have wished for better roommates (and above all, friends). Our dinners, late-night chats, "Campillo cinema" nights, the struggles to place sofas and tables in the living room in the best position to admire the sunsets together: all moments that will remain indelible in my heart. I wonder if one day I will learn how to put forks and knives in the dishwasher correctly. Perhaps it was worth earning the nickname "house of horrors"! I truly wish you all the best and a bright future. I sincerely believe in you.

Thanks to my parents for your unwavering trust, for being proud of me regardless of my choices, even when complicated, and for supporting me on every occasion. I hope I have never disappointed you and never will. Thanks to my sisters, Federica and Vittoria, and to Riccardo, for always being there at the right moments, despite the distance. Thanks to Benedetta for teaching me to recognise the truly important moments in life and not to overlook them. I hope I have been, and continue to be, a good "Uncle Gigi."

Thanks to Silvana, Santos, and Margherita for always making me feel like part of the family from the very beginning. Thank you for never doubting me and my 'extraordinary masonry' skills, and for the best life, literary, medical, and chess lessons I could have ever had.

Thanks to Girolamo, because friendship is not measured by the moments spent together. It is made of unbreakable bonds that neither time nor distance can weaken. I will never forget your constant support and your unwavering belief in me. Your words have always given me an indescribable energy. I hope I have been a good friend and will continue to be one in the future. I am grateful to you for having started it all by throwing the boards down the legendary lane 3, to announce your arrival at the pool. I cannot wait to finally organise our trip to London.

Thanks to all the volleyball teammates of 'Cuervos de San Nicasio'. With you, I learned that it doesn't matter whether we win or lose, but to enjoy every moment together, striving to improve step by step, regardless of the result.

A special thanks also to the lifelong friends from 'Unina San Juá' and '11s', who met in school or university classes, and who have remained friends long after those years. We grew up together, and this makes every achievement even more ours.

Finally, I want to thank Paola, the true cornerstone of my life. With you, I have truly learned the meaning of "bene velle." You have always been able to listen to me, understand me, and support me, even in the toughest choices or when everything seemed impossible. I wish everyone could have someone like you in their lives. We have been apart for some time, but I assure you that our sacrifices will have been worth it. Without you, none of this would have been possible. I dedicate this achievement to us, to our future adventures, our successes, and our dreams, certain that they are just within reach. There is still a long road ahead, together.

Ringraziamenti

Ed eccomi qui, dopo alcuni anni, davanti alla parte forse più delicata di questo percorso. Una sezione in cui non ci sono supervisori, né peer review, né Google Scholar da cui attingere riferimenti. Una sezione fatta solo di ricordi, emozioni e momenti che resteranno indelebili. Queste righe non appartengono solo a questa tesi, che segna la fine di un percorso, ma a un capitolo del libro della mia vita del quale sono immensamente orgoglioso. Non tanto per i traguardi raggiunti, quanto per le persone che mi hanno accompagnato lungo la strada e che spero continueranno a farlo. Il tempo trascorso con voi ha assunto una dimensione diversa, quasi inspiegabile scientificamente: questi anni sono volati, forse perché mi avete allargato la vita.

Voglio ringraziare innanzitutto le persone che hanno reso possibile questa avventura, iniziata ormai qualche anno fa. Grazie ai miei supervisori, Stefano e Andrea, per il supporto quotidiano, accademico e umano. Senza la vostra presenza costante, né questo dottorato né io stesso saremmo stati gli stessi. Ho sempre ammirato il vostro entusiasmo, la vostra perseveranza e l'energia con cui avete sempre cercato di fare di più, e di farlo meglio. Grazie per aver riconosciuto e valorizzato le mie potenzialità, ma soprattutto per avermi aiutato a conoscere i miei limiti e a superarli. Grazie per l'amicizia, per la fiducia e per avermi insegnato a guardare ogni traguardo come qualcosa di raggiungibile. Non avrei potuto desiderare maestri migliori, né dal punto di vista accademico né da quello umano.

Nella vita, in fondo, siamo sempre circondati da persone da cui imparare, da prendere come esempio. Tra queste voglio ringraziare Andrea I., Marco, Rodrigo e Andrea C. per essere stati per me un costante punto di riferimento scientifico e umano. Grazie per i momenti trascorsi insieme, dentro e fuori dall'università, e per aver sempre trovato il tempo per un consiglio o una parola giusta. È stando seduti al tavolo di un bar o in sella ad una bici che sono nate le migliori conversazioni. Non posso non menzionare, inoltre, Onofrio e Lionel. Grazie per la profondità dei vostri insegnamenti, per aver dato una svolta al mio percorso di dottorato e per avermi insegnato a guardare le cose da una prospettiva diversa. Grazie per l'entusiasmo e la passione per la scienza, ma anche per aver condiviso interessi che andavano ben oltre il lavoro. Spero di poter tornare presto, un giorno, ai caffè culturali delle pause trascorse con voi.

Grazie a tutte le persone che ho avuto la fortuna di incontrare all'Università.

Un ringraziamento particolare va a tutto il gruppo 'EAP Lab': a chi c'era prima e a chi c'è ora, dai dottorandi ai professori, fino ai program manager. Perché, in fondo, una famiglia è fatta anche dalle persone che scegliamo di avere accanto. Grazie di cuore per ogni momento condiviso, dalle discussioni scientifiche alle conferenze, fino agli eventi sociali più disparati. Un grazie speciale a Olympia per avermi guidato con pazienza tra le mille peripezie burocratiche. Grazie a Mr Chen, di poche parole, ma sempre pronto a sorprendere tutti con le sue idee brillanti.

Secondo alcuni studi, insultarsi a vicenda è un ingrediente fondamentale per un'amicizia sincera e duratura. Se così fosse, una delle persone con cui ho coltivato il legame più profondo è senza dubbio Iacopo, o più recentemente, "o' posdocc". Sei stata la prima persona che ho incontrato, ancora prima del mio trasferimento a Madrid, e se oggi sono arrivato fin qui lo devo in gran parte ai tuoi insegnamenti da fratello maggiore. Grazie per la complicità, per il sostegno e (talvolta, lo ammetto) per la pazienza, ma soprattutto per la tua costante presenza.

Insieme a Iacopo, non posso non citare Luca S., o "Scheli", come lo è sempre stato. Compagno di "grosselli" e "sushini", vero punto di riferimento di questa avventura. Se mi chiedessero di nominare i miei amici più sinceri, il tuo nome sarebbe sicuramente tra quelli in cima alla lista.

Grazie a Gabriele, un amico autentico, sempre pronto a offrire il consiglio giusto al momento giusto, compagno di campagne esplorative indimenticabili, dal Pantano estivo a Manzanares el Real (di cui forse valeva la pena visitare il castello!) e alle giornate di sci.

Grazie ad Enri, amico di nome, ma soprattutto di fatto. Non dimenticherò mai le nostre discussioni scientifiche in compagnia dei "ragazzi di Azkaban".

Grazie a tutti i membri dell' 'Oficina 1.2.A.06a', dagli albori con Patri e Alvaro fino ad Alicia, Qi, Pablo e Tani. Non dimenticherò mai i pranzi da Telepizza, le lunghe chiacchierate e le giornate intense di lavoro, rese più leggere da una colonna sonora improbabile tra villancicos, Pavarotti e rock. Grazie, Alicia, per avermi accompagnato costantemente in questo percorso, anche nei momenti più tortuosi, e per aver sopportato con pazienza ogni follia.

Grazie a Matteo R. per le infinite discussioni scientifiche e di cultura generale in Cercanías. Ho una profonda stima di te! Spero inoltre che un giorno sarò in grado di riuscire a murare una delle tue schiacciate a pallavolo.

Un grazie immenso va ai ragazzi di 'Plaza del Campillo del Mundo Nuevo 1', Luca, Donato (e Mich). Non avrei potuto desiderare coinquilini (e soprattutto amici) migliori. Le nostre cene, le chiacchierate fino a notte fonda, il "cinema Campillo", le lotte per posizionare divani e tavoli in salone nella posizione migliore per ammirare i tramonti insieme: tutti momenti che rimarranno indelebili nel mio cuore. Chissà se un giorno imparerò a mettere forchette e coltelli correttamente in lavastoviglie. Forse ne è valsa la pena guadagnarsi l'appellativo di "casa degli orrori"! Vi auguro davvero tutto il bene del mondo e un futuro splendido. Credo sinceramente in voi.

Grazie ai miei genitori, per aver sempre avuto fiducia in me ad occhi chiusi, per essere stati fieri di me indipendentemente dalle mie scelte, anche quando complicate, e per avermi supportato in ogni occasione. Spero di non avervi mai deluso e di non doverlo mai fare. Grazie alle mie sorelle, Federica e Vittoria, e a Riccardo, per esserci sempre stati nei momenti giusti, nonostante la lontananza. Grazie, Benedetta, per avermi insegnato a riconoscere i momenti davvero importanti della vita e a non trascurarli. Spero di essere stato, e di continuare a essere, un buon "zio Gigi".

Grazie a Silvana, Santos e Margherita per avermi sempre fatto sentire come parte della famiglia sin dall'inizio. Grazie per non aver mai dubitato di me e delle mie doti di 'straordinario muratore' e per le migliori lezioni di vita, letterarie, mediche e di scacchi che abbia mai potuto avere.

Grazie a Girolamo, perché l'amicizia non si misura con gli istanti trascorsi insieme. È fatta di legami indissolubili che né il tempo né la distanza possono scalfire o indebolire. Non dimenticherò mai il tuo supporto costante e il tuo incrollabile credere in me. Le tue parole mi hanno sempre trasmesso un'energia indescrivibile. Spero di essere stato un buon amico e di poterlo continuare a essere in futuro. Ti sono grato per aver iniziato tutto lanciando le tavolette dalla fantomatica corsia 3, per annunciare il tuo arrivo in piscina. Non vedo l'ora di organizzare finalmente il nostro viaggio a Londra.

Grazie a tutti i ragazzi della pallavolo di 'Cuervos de San Nicasio'. Con voi ho capito che non conta vincere o perdere, ma godersi ogni momento insieme, cercando di migliorare passo dopo passo, a prescindere dal risultato.

Un grazie speciale anche agli amici di sempre di 'Unina San Juá' e '11s', conosciuti tra i banchi di scuola o tra le aule universitarie, e rimasti amici ben oltre quegli anni. Siamo cresciuti insieme, e questo rende ogni traguardo ancora più nostro.

Infine vorrei ringraziare Paola, il vero pilastro portante della mia vita. Con te ho davvero imparato il senso del "bene velle". Sei sempre stata in grado di ascoltarmi, di capirmi e di supportarmi, anche nelle scelte più difficili o quando tutto sembrava impossibile. Auguro a tutti la presenza di una persona come te nella propria vita. Siamo ormai lontani da un po' di tempo, ma ti assicuro che questi nostri sacrifici saranno valsi a qualcosa. Senza di te tutto questo non sarebbe stato possibile. Dedico questo mio traguardo a noi, alle nostre future avventure, ai nostri successi e ai nostri sogni, sicurissimo che siano ad un passo dal realizzarsi. C'è ancora tanta strada da fare, insieme.

Data-driven modelling and predictive control of fluid flows

Luigi Marra
Department of Aerospace Engineering, Universidad Carlos III de Madrid
Avenida de la Universidad, 30, Leganés, Spain

Abstract

Fluid flows play a central role in many industrial processes, including transportation to energy production, thermal management, and chemical processing. The ability to actively manipulate these flows has long been recognised as a powerful means to optimise system performance. Examples of such applications include reducing aerodynamic drag in ground, airborne, and marine vehicles or friction in pipelines, improving efficiency in wind and hydrokinetic energy harvesting, and optimising heat transfer, combustion, and mixing processes, to name a few.

A fundamental prerequisite for effective flow control is the ability to predict their evolution over time, particularly under the influence of external inputs. However, this task is challenged by the inherent complexity of fluid flows, which feature high-dimensional state spaces, strong nonlinear interactions, intrinsic time delays, and multiscale behaviour across both spatial and temporal scales. These features make the identification of accurate and computationally tractable models a central difficulty in flow-control applications.

At the same time, the current data-rich era has created unprecedented opportunities. Advances in measurement technology, high-fidelity numerical simulations, and modern computing hardware have dramatically increased the availability of high-quality flow data. Parallel progress in machine learning and hardware has made it increasingly feasible to distil predictive dynamical models directly from such data. These developments have renewed interest in data-driven control and estimation strategies, particularly those requiring an explicit model of the system dynamics.

Among model-based approaches, Model Predictive Control (MPC) stands out as a particularly promising technique for flow control. MPC explicitly leverages a dynamical model to predict the future evolution of the system over a finite horizon and computes the optimal control action by repeatedly solving a constrained optimisation problem. Its ability to handle multivariable systems, incorporate state and input constraints, and exploit anticipatory behaviour makes it an attractive candidate for complex flow-control tasks.

Within this context, the present dissertation contributes to the development of control and data-driven model reduction for fluid flows. The first two contributions focus on flow control using MPC, while the third addresses the identification of low-dimensional coordinates suitable for estimation and control.

The first contribution proposes applying MPC to the control of aerodynamic forces in wake flows. Instead of modelling the full high-dimensional flow field, the approach identifies a low-dimensional model of the global aerodynamic forces acting on the body. This compact model is derived directly from data using the Sparse Identification of Nonlinear Dynamics framework, leveraging an assumption of compactness of the underlying governing laws. A key aspect of this work is the tuning of MPC hyperparameters, such as the prediction horizon and cost-function weights, which can be challenging in multi-objective optimisation.

To address this, an automatic tuning strategy based on Bayesian Optimisation is introduced, reducing user intervention while improving the controller performance. The robustness to disturbances and sensor noise is further enhanced through Local Polynomial Regression, providing more reliable state estimates for the MPC loop. The approach is evaluated on the fluidic pinball configuration, a system consisting of three cylinders arranged in an equilateral triangle and capable of independent rotation for actuation, at a Reynolds number equal to 150.

The second study investigates the applicability of MPC when the objective is not force control but rather direct manipulation of the flow field. The primary challenge in such settings is the intractable computational cost associated with formulating and solving an optimisation problem in a high-dimensional physical space. To overcome this difficulty, the proposed framework performs MPC in a low-dimensional latent space obtained through Proper Orthogonal Decomposition (POD). The POD identifies spatial modes capturing dominant variance patterns in the data, allowing the dynamics to be represented through a small number of modal coefficients. The system dynamics is then directly learned in this reduced order space using the data-driven non-intrusive Operator Inference technique. Another challenge addressed in this work is obtaining accurate state estimates from limited and noisy measurements. For this purpose, the proposed method integrates an Unscented Kalman Filter to reconstruct the latent-state representation, enabling the controller to reliably perform predictions and optimisations. The full closed-loop framework is demonstrated on one- and two-dimensional versions of the Kuramoto–Sivashinsky equation, a canonical example of complex, nonlinear, spatially extended dynamics.

Finally, the last contribution addresses the challenge of constructing reduced-order models that identify coordinates relevant for controlling complex fluid flows. Building on the manifold hypothesis, which states that high-dimensional flow data often lie on a low-dimensional manifold, the work applies the Isometric Mapping (ISOMAP) algorithm to uncover intrinsic coordinates of the post-transient wake dynamics of the fluidic pinball at Reynolds number equal to 30 under actuation. ISOMAP identifies a compact five-dimensional representation: two coordinates capture downstream vortex-shedding oscillations, while three describe the near-field actuation response. This representation reduces dimensionality while providing a physically meaningful decomposition of the flow dynamics. The method is further extended with a deterministic decoding strategy, using k-Nearest Neighbours and neural networks, to reconstruct flow snapshots from sparse sensors and actuation data with low error. Even though a predictive model is not directly identified, this approach sets a fundamental step towards manifold-based reduced-order models suitable for real-time control.

The investigations presented in this dissertation contribute to the development of MPC for complex fluid flows and demonstrate the potential of data-driven modelling techniques to make such applications computationally feasible. The results highlight the promise of combining MPC with modern data-driven identification strategies, paving the way toward efficient and scalable control frameworks for high-dimensional, nonlinear flows.

Key words: Data-Driven System Identification, Reduced-Order Model, Model Predictive Control, Manifold Learning, Optimisation

Preface

This thesis focuses on the application of data-driven modelling strategies and their subsequent use in predictive control for fluid dynamics problems. It includes three journal publications, which have been reformatted to respect the style of this document without altering the scientific content or the intended message of each publication.

March 2026, Leganés
Luigi Marra

Published and submitted content

The following publications are fully included in Part II of this thesis:

Paper 1. Marra, L., Meilán-Vila, A., & Discetti, S. 2024. Self-tuning model predictive control for wake flows. *J. Fluid Mech.*, 983, A26. DOI: 10.1017/jfm.2024.47.

Paper 2. Marra, L., Semeraro, O., Mathelin, L., Meilán-Vila, A., & Discetti, S. 2025. Latent-space non-linear model predictive control for partially-observable systems. *arXiv preprint.* DOI: arXiv.2511.19056. (Under review)

Paper 3. Marra, L., Cornejo Maceda, G. Y., Meilán-Vila, A., Guerrero, V., Rashwan, S., Noack, B. R., Discetti, S. & Ianiro, A. 2024. Actuation manifold from snapshot data. *J. Fluid Mech.*, 996, A26. DOI: 10.1017/jfm.2024.593.

Whenever material from this source is included in this thesis, it is singled out with typographic means and an explicit reference.

Division of work between authors

The description of the individual author contributions is presented by means of CRediT. CRediT (Contributor Roles Taxonomy) provides an accurate and detailed description of the authors' diverse contributions to the published work.

Paper 1.
LM: Data curation; Formal analysis; Investigation; Methodology; Software; Validation; Visualization; Writing – original draft. **AMV**: Conceptualisation; Funding acquisition; Methodology; Project administration; Resources; Supervision; Writing - review & editing. **SD**: Conceptualisation; Funding acquisition; Methodology; Project administration; Resources; Supervision; Writing - review & editing.

Paper 2.
LM: Conceptualization; Data curation; Formal analysis; Funding acquisition; Investigation; Methodology; Project administration; Resources; Software; Validation; Visualization; Writing - original draft. **OS**: Conceptualization; Funding acquisition; Investigation; Methodology; Resources; Supervision; Writing - review & editing. **LiM**: Conceptualization; Investigation; Methodology; Resources; Supervision; Writing - review & editing. **AMV**: Conceptualization; Funding acquisition; Investigation; Methodology; Resources; Supervision; Writing - review & editing. **SD**: Conceptualization; Formal analysis; Funding acquisition; Investigation; Methodology; Project administration; Resources; Supervision; Writing - review & editing.

Paper 3.
LM: Data curation; Formal analysis; Investigation; Methodology; Software; Validation; Visualization; Writing - original draft; Writing – review & editing. **GCM:** Formal analysis; Methodology; Resources; Validation; Visualization; Writing - review & editing. **AMV** Funding acquisition; Methodology; Project administration; Supervision; Writing - review & editing. **VG:** Funding acquisition; Project administration; Resources; Supervision; Writing - review & editing. **SR:** Data curation; Investigation; Methodology; Software;

Formal analysis. **BN:** Conceptualization; Funding acquisition; Methodology; Project administration; Resources; Supervision; Writing – original draft; Writing – review & editing. **SD:** Conceptualization; Funding acquisition; Methodology; Project administration; Supervision; Writing - review & editing. **AI:** Conceptualization; Funding acquisition; Methodology; Project administration; Supervision; Writing - review & editing.

Acronyms for Author and Co-Authors:

LM	Luigi Marra	Universidad Carlos III de Madrid, (*PhD Candidate*)
SD	Stefano Discetti	Universidad Carlos III de Madrid, (*Advisor and Tutor*)
AMV	Andrea Meilán-Vila	Universidad Carlos III de Madrid, (*Advisor*)
AI	Andrea Ianiro	Universidad Carlos III de Madrid
OS	Onofrio Semeraro	CNRS – Université Paris-Saclay
LiM	Lionel Mathelin	CNRS – Université Paris-Saclay
BN	Bernd R. Noack	Harbin Institute of Technology
GCM	Guy Y. Cornejo Maceda	Harbin Institute of Technology
VG	Vanesa Guerrero	Universidad Carlos III de Madrid
SR	Salma Rashwan	Universidad Carlos III de Madrid

Other research merits

In addition to the three publications listed above, the following works have also been published as part of the research activity:

Co-authored article

Hou, C., Marra, L., Cornejo Maceda, G. Y., Jiang, P., Chen, J., Liu, Y., Hu, G., Chen, J., Ianiro, A., Discetti, S., Meilán-Vila, A. & Noack, B. R. 2024. Machine-learned flow estimation with sparse data—Exemplified for the rooftop of an unmanned aerial vehicle vertiport. *Phys. Fluids*, 36 (12), 125198. DOI: 10.1063/5.0242007.

Co-authored book chapters

Discetti, S., Marra, L. & Meilán-Vila, A. 2025, Model Predictive Control. In Mendez, M. A. & A. Parente (Eds.), Machine Learning for Fluid Dynamics (pp. 231–251). Von Karman Institute for Fluid Dynamics. ISBN: 978-2875162090.

Bulletin articles

Marra, L., Meilán-Vila, A. & Discetti, S. 2025. Self-tuning model predictive control for unsteady aerodynamic force tracking in wake flows. *ERCOFTAC Bulletin*, 144, 39–44. DOI: 10.57651/b144.8.

Conferences

Part of the work in this thesis has been presented at the following international conferences. The presenting author is underlined.

<u>Marra, L.</u>, Semeraro, O., Mathelin, L., Meilán-Vila, A., & Discetti, S., *Efficient control of dynamical systems via Model Predictive Control and Reinforcement Learning.* Euromech Colloquium on Data-Driven Fluid Dynamics & 2nd ERCOFTAC Workshop on Machine Learning for Fluid Dynamics. London, UK, 2025.

<u>Marra, L.</u>, Cornejo Maceda G. Y., Meilán-Vila, A., Guerrero, V., Rashwan, S., Noack, B. R., Discetti, S., Ianiro, A., *Manifold learning from snapshot data.* APS Division of Fluid Dynamics Meeting 2024. Salt Lake City, US, 2024.

Marra, L., Meilán-Vila, A., & <u>Discetti, S.</u>, *Drag reduction with self-tuning model predictive control.* European Drag Reduction and Flow Control Meeting – ERCOFTAC. Turin, Italy, 2024.

<u>Marra, L.</u>, Meilán-Vila, A., & Discetti, S., *Automatic parameter selection for model predictive control for fluid flows.* 2nd Spanish Fluid Mechanics Conference. Barcelona, Spain, 2023.

<u>Marra, L.</u>, Meilán-Vila, A., & Discetti, S., *Kernel-based Model Predictive Control for Fluid Flows in the Presence of Noise.* 1st Joint Workshop on Functional Data Analysis and Nonparametric Statistics. Madrid, Spain, 2023.

Marra, L., Meilán-Vila, A., & Discetti, S., *Self-tuning model predictive control for fluid flows.* 1st International Conference Math2Product (M2P 2023). Taormina, Italy, 2023.

Marra, L., Meilán-Vila, A., & Discetti, S., *Effect of measurement noise on MPC for drag reduction.* 1st International Conference on Mathematical Modelling in Mechanics and Engineering. Belgrade, Serbia, 2022.

International mobility

A 90-day research stay (September–December 2024) was carried out at the *Laboratoire Interdisciplinaire des Sciences du Numérique (LISN), CNRS – Université Paris-Saclay*, under the supervision of professors Onofrio Semeraro and Lionel Mathelin. This research stay was focused on the study and development of hybrid approaches involving model predictive control and reinforcement learning. This stay was supported by the aid *Programa propio de investigación y transferencia, para la movilidad de investigadores/as de la Universidad Carlos III de Madrid en centros de investigación nacionales o extranjeros.*

Dissemination activities

13/11/2025: Madrid Science and Innovation Week 2025, *Ciencia del empuje en acción: cómo funciona un motor de avión.* Universidad Carlos III de Madrid.

12/11/2025: Madrid Science and Innovation Week 2025, *Todos somos entrenadores: descubre cómo aprende una inteligencia artificial.* Universidad Carlos III de Madrid.

12/06/2024: Thesis Talk Event 2024, *Aprendiendo de la naturaleza a controlar la turbulencia.* Universidad Carlos III de Madrid.

06/04/2024: Teatro y divulgación científica: navegando los retos de una Europa verde, *Aprende a controlar la aerodinámica.* Universidad Carlos III de Madrid.

09/01/2024: STEM FOR GIRLS Workshops 2024, *Descubre el encanto de controlar el agua.* Universidad Carlos III de Madrid.

15/11/2023: Madrid Science and Innovation Week 2023, *Descubre la magia de controlar el aire o el agua.* Universidad Carlos III de Madrid.

15/06/2023: Thesis Talk Event 2023, *Controlling fluids as in the game of chess.* Universidad Carlos III de Madrid.

Funding

This work was supported (Dec 2023 – Mar 2026) by the funding under 'Orden 3789/2022, del Vicepresidente, Consejero de Educación y Universidades, por la que se convocan ayudas para la contratación de personal investigador predoctoral en formación para el año 2022' and by the research project PREDATOR-CM-UC3M (Oct 2022 – Nov 2023), funded by the call "Estímulo a la Investigación de Jóvenes Doctores/as" within the frame of the Convenio Plurianual CM-UC3M and the V PRICIT (V Regional Plan for Scientific Research and Technological Innovation).

Contents

List of Acronyms

BO	Bayesian Optimisation
DA	Data Assimilation
DeepMPC	Deep Model Predictive Control
DNS	Direct Numerical Simulation
DP	Dynamic Programming
HJB	Hamilton–Jacobi–Bellman
ISOMAP	Isometric Mapping
KF	Kalman Filter
kNN	k-Nearest Neighbours
KS	Kuramoto–Sivashinsky
LQG	Linear Quadratic Gaussian
LQR	Linear Quadratic Regulator
LSTM	Long Short-Term Memory
MDS	Multidimensional Scaling
MPC	Model Predictive Control
OpInf	Operator Inference
POD	Proper Orthogonal Decomposition
RL	Reinforcement Learning
ROM	Reduced-Order Model
SINDy	Sparse Identification of Non-linear Dynamics
UKF	Unscented Kalman Filter

Part I

Overview and summary

Chapter 1

Introduction

This chapter presents the motivation and objectives of the work. It first outlines the historical and industrial context, highlighting the importance of fluid flow control, and then details the core research goals and specific sub-objectives. Finally, it summarises the thesis structure and the original research contributions presented in the included papers.

1.1. Motivation

Control is the ability to manipulate a given system to achieve a desired objective. When applied to fluid flows, this endeavour traces its roots back through the annals of history, representing one of the oldest and most persistent engineering challenges of humanity. Indeed, even in the prehistoric era, humans displayed a nascent understanding of flow control (Gad-el Hak & Pollard, 1998), designing stabilising fins for arrows or aerodynamically shaped spears for hunting, instinctively leveraging principles of aerodynamics. With the rise of the first great civilisations, this ingenuity scaled dramatically. The development of aqueducts and embankments for irrigation represented one of the earliest large-scale and intentional efforts to regulate and direct water flow. Likewise, the deliberate optimisation of hull geometries to improve speed and efficiency at sea reflected a growing command of hydrodynamic behaviour. A pioneering moment in modern flow control came on December 17, 1903, with the first sustained, manned flight of the Wright brothers (McCullough, 2015). By successfully manipulating moving control surfaces, they demonstrated for the first time the efficacy of active control of an aircraft, holding a flight for approximately 12 seconds. This proof-of-concept would fundamentally alter transportation. The transition from the *engineering era* to the *scientific era* of flow control was heralded in 1904, when Ludwig Prandtl introduced the seminal theory of the Boundary Layer and its control (Prandtl, 1904). Since then, the understanding of fluid flows has accelerated significantly, building upon the foundational studies of pioneers like Osborne Reynolds (Reynolds, 1883), allowing us to describe their complex laminar and turbulent behaviours.

The same motivations that historically drove the dedication to fluid flow control remain at the core of countless modern applications. Fluid flows are ubiquitous in industrial and scientific domains: from transportation and energy production to chemical and food processing, and critical areas of medicine (see Figure 1.1). In all these sectors, flow control emerges as a key enabler for goals such as: drag reduction in marine, automotive and aerial vehicles; control of efficiency and load reduction in modern wind turbines; optimisation of mixing and heat transfer in refinery and chemical processing; and precise fluid manipulation in microfluidic and biomedical systems.

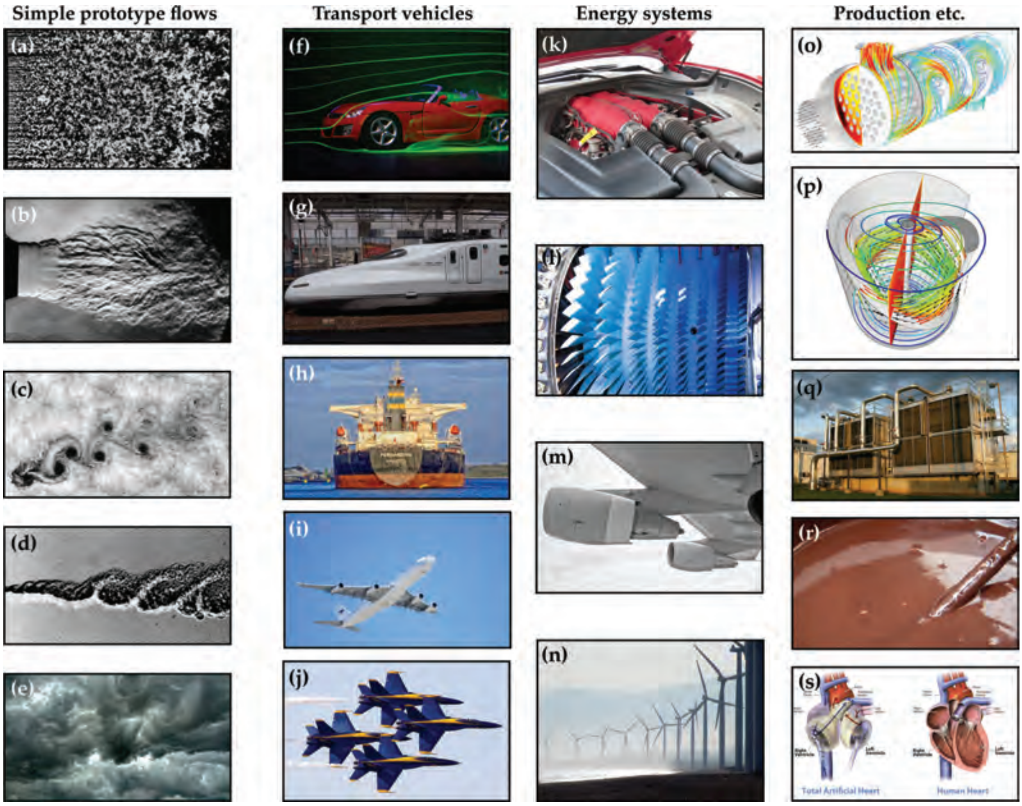

Figure 1.1: Applications of fluid flow control across multiple domains; from fundamental prototype flows to real-world systems. Panels include simple laboratory-scale flows, transport vehicles, energy-related systems and process industry. Figure reproduced with permission from Brunton & Noack (2015).

To appreciate the sheer scale of the role of fluids, approximately 25% of the energy consumed in industry and commerce is dedicated solely to moving fluids through pipes and channels (Jiménez, 2013). The impact on the transport sector is equally massive, accounting for roughly 20% of global energy consumption (Brunton & Noack, 2015). This translates into a significant environmental burden, as road transport alone surpassed 22×10^9 tonnes in CO_2 emissions in the decades leading up to 2015, with projections expecting a 57% increase by 2030. A control action yielding even a negligible percentage of improvement in these areas could have a significant effect. To give a practical example in the automotive sector, a large portion of emissions is due to aerodynamic drag, accounting for 50% or even 80% depending on speed (Brunton & Noack, 2015). Flow control can contribute to as much as approximately 25% drag reduction. This is particularly salient given the ambitious global targets, such as the goal of the aviation sector to achieve net-zero carbon emissions by 2050 (Air Transport Action Group (ATAG), 2025; International Air Transport Association (IATA), 2025). Furthermore, the motivation for developing innovative flow control techniques lies in their cross-cutting impact, yielding results that are valuable across multiple industrial sectors.

However, controlling fluid flows is an inherently complex task. This difficulty stems from their defining nature: their behaviour is highly non-linear, high-dimensional, multiscale, and often characterised by significant time delays (Duriez *et al.*, 2017). This complexity has led to the development of a wide range of flow control strategies. Early approaches focused on passive control, based on fixed geometrical modifications, while later developments introduced active control, in which energy or momentum is injected into the flow (Gad-el Hak, 1996). In this last category, control strategies have been formulated either in an open-loop fashion, where actuation is prescribed a priori, or in a closed-loop configuration, where control actions depend on measurements of the evolving flow state. Among these, feedback-based closed-loop control has emerged as a promising paradigm, as control actions are adjusted in real time based on measurements of the evolving flow state, enabling robustness to disturbances and modelling uncertainties.

A key enabler of feedback-based control is the availability of a dynamical model capable of predicting the response of the flow to control actions, which is essential for planning and optimising control decisions. In principle, such a model is provided by the Navier-Stokes equations, which govern the flow dynamics and describe the system behaviour in time, even in the presence of external actuation. However, despite their completeness, the direct use of the Navier-Stokes equations in real-time estimation and control applications is practically infeasible. An analytical solution is generally elusive, and a numerical solution requires fine spatial and temporal discretisation. This results in state spaces of extremely high dimensionality, with computational costs that scale unfavourably with the Reynolds number, rendering real-time estimation and control practically infeasible for almost all applications.

This is where a paradigm shift is occurring (Tolle *et al.*, 2011). Significant advancements in hardware and technology, specifically in high-performance computing, numerical algorithms, optimisation, simulation, and Machine Learning, are converging to create unprecedented opportunities. This convergence contributes to two fundamental aspects: an ever-increasing availability of high-fidelity data harvested from both numerical simulations and real experiments, and the development of sophisticated algorithms capable of distilling Reduced-Order Models (ROM) that describe the essential, lower-dimensional behaviour of fluid flows (Lassila *et al.*, 2014). Considerable effort is now being placed on identifying these reduced models, which exploit the property of turbulent flows to evolve on an attractor of intrinsically low dimensionality. This is motivated especially by the presence of coherent structures within the flow, such as vortices or shear layers, which govern the dominant transport mechanisms (Hussain, 1986; Robinson, 1991) and can be effectively captured by a small number of features. In this context, data should be interpreted as representations of empirical evidence rather than as an end in themselves, with the central challenge lying in the informed use of physical insight and efficient algorithms to distil the underlying, simplified structure that governs the observed phenomena.

The successful identification of efficient, predictive models has made advanced control strategies for fluid systems a practical reality. Accurate, low-dimensional models are particularly advantageous for optimisation-based approaches that rely on the ability to project the system evolution into the future. Such algorithms exploit the predictive capability of a model to evaluate the consequences of different control actions, allowing the selection of an optimal control sequence while respecting physical and operational constraints. This predictive, forward-looking perspective is especially critical in fluid flows, where delays, nonlinearities, and multiscale interactions make real-time decision-making challenging. This paradigm, leveraging a model to predict the future evolution of the system, is precisely what is exploited by Model Predictive Control (MPC, Camacho & Bordons, 2007). It is a control strategy that uses a model to continuously predict the future behaviour of the system over a finite horizon and determines control actions by solving an optimisation problem based on these predictions. It was originally developed for industrial processes (Garcia *et al.*, 1989), particularly those in which system constraints must be explicitly enforced.

The development of MPC for fluid flows is exceptionally promising, often offering a more robust framework than other state-of-the-art strategies. Its primary advantage is the ability to anticipate future behaviour through a predictive model. Beyond this, MPC explicitly incorporates constraints on states and actuators within the control design, a critical aspect for fluid systems subject to limited actuation authority or energetic bounds.

However, to move from theoretical promise to industrial utility, the application of MPC in this field faces fundamental challenges. The accuracy of the control relies entirely on the underlying ROM, making the extraction of governing dynamics for an effective ROM a non-trivial task, particularly when dealing with actuated flows. This requires the development of robust data compression techniques to handle and process the massive datasets before model identification can even begin. Furthermore, for these controllers to operate reliably, they must be robust, adaptive, and capable of operating under real-world conditions, including significant noise and varying flow regimes. Finally, a fundamental practical limitation in closed-loop control is the estimation problem: in physical systems, we rarely have access to the entire, high-dimensional state of the flow, relying instead on sparse, localised sensor measurements. Therefore, any viable control strategy requires highly accurate and computationally light state estimation techniques to infer the full flow state from limited data, bridging the gap between measurements and the requirements of the model.

1.2. Objectives

The primary objective of this dissertation is to develop data-driven frameworks for modelling and control of fluid flows with predictive strategies under conditions of limited observability and uncertainties.

This research addresses the fundamental challenges of predictive control in fluid flows and benchmark systems exhibiting similar dynamics, characterised by strong nonlinearity, multiscale interactions, and chaotic behaviour. The work focuses on the identification of compact, data-driven surrogate models of the dynamics and their use within predictive control frameworks. MPC is employed as the core control strategy, while data-driven methods provide the tools for model reduction, dynamic compression, and surrogate model identification.

Within this framework, some specific sub-objectives are identified:

- Enable accurate reduced-order modelling for control through suitable low-dimensional representations of the flow
- Build predictive models of the flow dynamics using machine-learning-based system identification

- Develop and implement MPC strategies based on the identified models, ensuring optimal control action over a finite horizon.
- Develop and integrate data-driven state estimation techniques
- Include in the framework robustness to typical sources of uncertainty and implementation constraints, such as measurement noise, model-plant mismatch, sparse sensing, and automatic controller tuning.
- Systematically validate the proposed framework on fluid flow systems and on canonical benchmarks exhibiting similar relevant properties.

These objectives collectively contribute to assessing the tractability of MPC and advancing its practical implementation, helping to transform it into a deployable technology for real-time control of increasingly fluid flow applications.

1.3. Thesis structure

The dissertation is organised into two main parts. Part I presents the theoretical background underpinning this dissertation. The rest of Part I is organised as follows. Chapter 2 introduces the foundations of Model-Based Control, covering the formulation of Optimal Control and outlining the principles of Approximate Dynamic Programming and Adaptive Control. The chapter also discusses the specific challenges arising in the modelling of turbulent flows and reviews the most relevant data-driven identification techniques for this thesis. Chapter 3 provides a comprehensive discussion of MPC. It introduces the backbone and motivations behind MPC, explains its algorithmic structure and practical implementation, and reviews its main applications to fluid flow control. The chapter then examines recent developments that integrate learning mechanisms within the MPC framework and concludes by highlighting the links between MPC and Reinforcement Learning (RL). Chapter 4 describes the two test cases used throughout the thesis, the Fluidic Pinball and the Kuramoto–Sivashinsky (KS) equation, detailing their governing equations, dynamical features, and aspects relevant to control. Finally, Chapter 5 summarises the main contributions of the thesis and outlines directions for future research.

Part II presents the three original research contributions developed during this doctoral work. Specifically, the first article (**Paper 1**) introduces an MPC-based strategy for wake flow control. The predictive model is obtained in a fully data-driven manner, using an identification procedure that promotes sparsity. The work focuses on identifying a completely self-tuning controller, requiring minimal user intervention thanks to an external hyperparameter optimisation. Robustness to measurement noise is ensured by Local Polynomial Regression, which provides reliable estimates of forces and their time derivatives directly from data.

Paper 2 addresses the application of MPC to high-dimensional fluid flow systems in the presence of sparse and noisy measurements. The control is defined directly in a latent space to mitigate the challenges of high dimensionality, and a state estimator is employed to reconstruct the system feedback from noisy observations, providing an approach suitable for practical operational conditions.

Finally, **Paper 3** focuses on reduced-order modelling for controlled fluid flows. It introduces a methodology to encode snapshots of the controlled flow field based on the manifold hypothesis, coupled with a decoding strategy using neural networks and k-Nearest Neighbours to minimise reconstruction error. This approach provides low-dimensional representations with improved reconstruction and estimation accuracy, even when only a limited number of flow sensors are available.

The aforementioned papers are listed below:

Paper 1. Marra, L., Meilán-Vila, A., & Discetti, S. 2024. Self-tuning model predictive control for wake flows. *J. Fluid Mech.*, 983, A26. DOI: 10.1017/jfm.2024.47.

Paper 2. Marra, L., Semeraro, O., Mathelin, L., Meilán-Vila, A., & Discetti, S. 2025. Latent-space non-linear model predictive control for partially-observable systems. *arXiv preprint*. DOI: arXiv.2511.19056. (Under review)

Paper 3. Marra, L., Cornejo Maceda, G. Y., Meilán-Vila, A., Guerrero, V., Rashwan, S., Noack, B. R., Discetti, S. & Ianiro, A. 2024. Actuation manifold from snapshot data. *J. Fluid Mech.*, 996, A26. DOI: 10.1017/jfm.2024.593.

Chapter 2

Model-Based Control

This chapter begins by providing a formal definition of the control problem and then introduces the model-based paradigm, where an explicit representation of the system dynamics forms the foundation for the controller design. The theoretical principles of model-based control are presented alongside key contributions from the literature. The main results for both linear and nonlinear systems are discussed, and the fundamentals of optimal control are introduced. Approximate dynamic programming methods are then presented, highlighting their utility in situations where exact solutions are computationally intractable. The chapter then addresses the primary challenges of model identification in fluid dynamics, emphasising how the high dimensionality and intrinsic complexity of turbulent flows require dimensionality reduction techniques. Finally, an overview of the system identification methods most pertinent to this dissertation is provided.

2.1. Formulation of the control problem

At any continuous time instant t, the state of a dynamical system can be represented by a vector $\boldsymbol{x}(t) \in \mathbb{R}^{n_x}$, where n_x corresponds to the number of degrees of freedom. The system can be influenced through an input vector $\boldsymbol{u}(t) \in \mathbb{R}^{n_u}$. The time evolution of the system is described by the differential equation

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\nu}_x), \quad \boldsymbol{x}(0) = \boldsymbol{x}_0, \tag{2.1}$$

where $\dot{\boldsymbol{x}}$ denotes the time derivative of the state vector, $\boldsymbol{\nu}_x \in \mathbb{R}^{n_{\nu_x}}$ is a vector of external disturbances, $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_{\nu_x}} \to \mathbb{R}^{n_x}$ defines the system's temporal evolution, and $\boldsymbol{x}_0$ is the initial state. It is worth noting that, more generally, the system dynamics may also have an explicit dependence on time. However, in this dissertation, the focus is on *autonomous* systems, i.e. with governing equations that do not depend explicitly on time. Some examples of disturbances $\boldsymbol{\nu}_x$ include wind gusts, environmental vibrations, and actuator imperfections, for instance. In some cases, these disturbances can be modelled as additive, i.e., entering linearly into the state evolution.

The control problem can also be formulated in a discrete-time formulation. Let the discretisation of a generic continuous variable $\boldsymbol{x}(t)$ be denoted by $\boldsymbol{x}_k = \boldsymbol{x}(t_k)$, where $t_k = k\Delta t$, $k = 1, 2, \dots$ with constant sampling interval Δt. Under this assumption, the discrete dynamical system becomes:

$$\boldsymbol{x}_{k+1} = F(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\nu}_{x_k}). \tag{2.2}$$

where F is the discrete-time evolution map that approximates the effect of the continuous-time dynamics over one sampling interval.

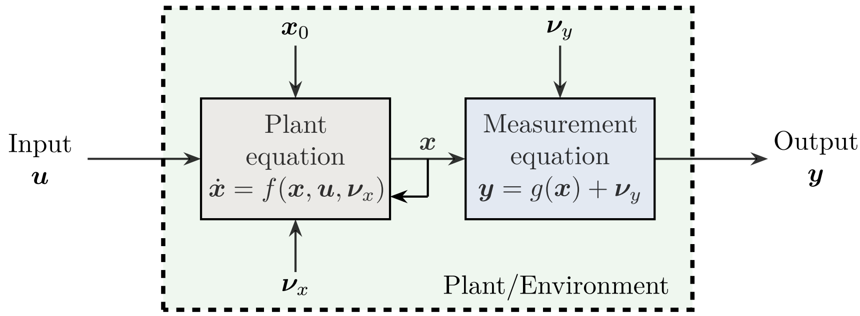

Figure 2.1: Schematic representation of the plant. The system state $\boldsymbol{x}$ evolves according to the dynamics f, influenced by the input $\boldsymbol{u}$ and external disturbances $\boldsymbol{\nu}_x$. Measurements $\boldsymbol{y}$ are obtained via sensors through the measurement function g, possibly corrupted by measurement noise $\boldsymbol{\nu}_y$.

In flow control applications, the system state $\boldsymbol{x}(t)$ can represent the full flow field information, such as velocity and pressure at all points resulting from a discretisation, or it can correspond to a reduced or latent representation of the flow capturing the essential dynamics. In some cases, it may also consist of global or proxy quantities derived from the flow, such as aerodynamic forces (e.g., drag and lift). These states are often not directly measurable, and the available feedback typically comes from measurements obtained with probes (e.g., hot-wire velocity or pressure sensors) or particle image velocimetry in selected regions, due to sensor limitations and the need for fast data acquisition. Consequently, only partial observations of the system are available, which are collected through a set of sensors and formalised as measured outputs:

$$\boldsymbol{y} = g(\boldsymbol{x}) + \boldsymbol{\nu}_y \in \mathbb{R}^{n_y}, \tag{2.3}$$

where n_y is the number of measured outputs, $g : \mathbb{R}^{n_x} \to \mathbb{R}^{n_y}$ is the measurement function, and $\boldsymbol{\nu}_y \in \mathbb{R}^{n_y}$ represents additive measurement disturbances due to measurement noise. The Equations (2.1) and (2.3) model the process of interest for the control problem. In control terminology, this is generally referred to as the *plant*, while in Dynamic Programming (DP) or RL contexts, it is called the *environment.* A graphical representation of the plant is shown in Figure 2.1. The figure highlights the black box nature of the interpretation of the plant dynamics, which can be seen as an input-output relation. The inputs are the actions affecting the system dynamics, while the outputs are the measurements of the system.

The goal of control is to determine the input signal $\boldsymbol{u}(t)$ that steers the system toward trajectories or states associated with desirable outcomes, typically using feedback information obtained from the measurements $\boldsymbol{y}(t)$. Depending on how the control inputs are generated, two broad categories of control strategies can be distinguished. *Open-loop control* determines the control input based solely on a predefined plan, without incorporating information from the current state or output measurements. In contrast, *closed-loop control* (or feedback control) continuously adjusts the input $\boldsymbol{u}(t)$ based on real-time measurements. The feedback signal, typically the measured output $\boldsymbol{y}(t)$, or an estimated state derived from it, encapsulates information about the current behaviour of the system. Formally, the expected control in the closed loop scenario law assumes the form

$$\boldsymbol{u} = \mathbf{H}(\boldsymbol{y}) \tag{2.4}$$

where $\mathbf{H} : \mathbb{R}^{n_y} \to \mathbb{R}^{n_u}$ denotes the mapping from the measured output to the control input obtained according to the specific control target.

In this dissertation, the focus is exclusively on closed-loop control strategies, as feedback enables the controller to estimate the system state and adapt control actions in real time, thereby improving performance. The desired system behaviour is formalised through a general performance measure to be optimised. Depending on the objective, this may involve stabilising an unstable system, attenuating sensor noise, compensating for exogenous disturbances (Duriez *et al.*, 2017). In many applications, this control procedure must satisfy physical, safety, or operational constraints. These are expressed as admissible sets for states and inputs $\mathcal{X} \subseteq \mathbb{R}^{n_x}$, $\mathcal{U} \subseteq \mathbb{R}^{n_u}$, which define the regions in which states and inputs must remain during operation. In their simplest form, these sets can be expressed using linear inequalities:

$$\mathcal{X} = \{\boldsymbol{x} \in \mathbb{R}^{n_x} \mid \mathbf{F}_x \boldsymbol{x} \leq \boldsymbol{g}_x\}, \qquad \mathcal{U} = \{\boldsymbol{u} \in \mathbb{R}^{n_u} \mid \mathbf{F}_u \boldsymbol{u} \leq \boldsymbol{g}_u\}, \tag{2.5}$$

where $\mathbf{F}_x, \mathbf{F}_u$ and $\boldsymbol{g}_x, \boldsymbol{g}_u$ are matrices and vectors, respectively, of appropriate dimensions defining the bounds on states and inputs.

2.2. Principles of Model-Based Control

A model is a mathematical representation of the equations describing the evolution of the plant to be controlled, as shown in Equation (2.1). The model must be capable of simulating the input-output behaviour of the plant in terms of state transition, under the influence of control inputs or external disturbances. Moreover, it must reproduce, where required, the measurements of the states.

Model-Based Control refers to any solution of the control problem introduced in Section 2.1 that explicitly utilises a plant model to compute the control actions. This term encompasses an entire class of algorithms that employ a plant model for control applications. Even though the usage of models in control and estimation problems was not new already in the 1950s, the first notable identification of a model for *state and state transition* dates back to the pioneering work of Kalman (1960), who demonstrated the utility of a linear state-space model for recursive estimation. Since then, model-based control has been applied to a wide range of systems, from linear to nonlinear, forming the foundation of modern control theory and a diverse set of algorithmic frameworks (Andrei, 2006).

The primary advantage of model-based approaches lies in their planning and prediction capability. The use of a model enables planning over future time horizons: the model can be used to simulate or predict the evolution of the system under candidate control input sequences, and control actions can be selected to optimise a prescribed performance criterion over these trajectories. The effect of different actions can thus be assessed before they are implemented in the plant for control. This is particularly relevant for control tasks requiring constraint handling or optimality. In contrast, model-free approaches do not rely on an explicit system model (Hou & Wang, 2013). They compute control actions directly from measurements or observed outputs, without simulating future states. While model-free methods can offer flexibility when accurate models are unavailable, model-based control provides the key advantage that control laws can be derived directly from a causal description of the plant dynamics, even though this generally comes at the expense of an increased computational cost.

Plant models can be derived from various sources. They may be obtained from first-principles physics, often requiring simplification of complex governing equations, or they can be built directly from data using system identification techniques. While having a high-fidelity model is crucial for effective model-based control, errors in the model are unavoidable, especially in complex systems like turbulent flows. These might originate from non-modelled nonlinearities, parameter uncertainty, or simplifications introduced to make the model computationally tractable. These limitations have prompted the creation of robust control strategies that explicitly consider model uncertainty, improving operation when the plant dynamics are only roughly known (Dorato, 2003; Lin, 2007).

The following discussion will focus on the principal results of model-based control, beginning with optimal control in both linear and nonlinear systems, and subsequently addressing the challenges associated with modelling plant dynamics in complex systems such as fluid flows. The discussion will also review the modelling and system-identification techniques employed in this dissertation to construct tractable and predictive models for control purposes.

2.3. Optimal Control

A model of the plant dynamics opens the possibility of solving the control problem in an optimal sense, leading to the vast theory of Optimal Control. It can be interpreted as a mathematical formulation of the control problem introduced in Section 1.1, where the objective is to optimise a well-defined measure of system performance (Sethi, 2021). To this end, a state- and input-dependent cost function is defined:

$$c : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}, \qquad c \equiv c(\boldsymbol{x}, \boldsymbol{u}), \tag{2.6}$$

which quantifies the instantaneous penalty of being in state $\boldsymbol{x}$ and applying the control input $\boldsymbol{u}$. Based on this cost function definition, the cost-to-go, also known as performance index, for a given control $\boldsymbol{u}$ is

$$J(\boldsymbol{u}) = \int_t^T c(\boldsymbol{x}(\tau), \boldsymbol{u}(\tau))\, \mathrm{d}\tau + h(\boldsymbol{x}(T)), \tag{2.7}$$

where $h : \mathbb{R}^{n_x} \to \mathbb{R}$ is a terminal cost function penalising the state at the final time $t = T$. The terminal cost can be used to approximate the cost-to-go beyond the finite horizon or to impose specific final-state constraints. In model-based control, the evaluation of the cost functional is made possible by the availability of the model of the system dynamics capable of predicting the evolution of the system state $\boldsymbol{x}(t)$ under the selection of the input $\boldsymbol{u}(t)$.

The continuous-time optimal control problem is then formulated as

$$\boldsymbol{u}^\star = \arg\min_{\boldsymbol{u} \in \mathcal{U}} J(\boldsymbol{u}), \tag{2.8}$$

subject to the system dynamics represented by the plant model. When T is finite, the problem is referred to as a finite-horizon optimal control problem, where the terminal cost $h(\boldsymbol{x}(T))$ plays an important role. In the infinite-horizon case, where $T \to \infty$, the terminal cost is typically omitted, and additional considerations are necessary to ensure the convergence of the integral in Equation (2.7), introducing a discounting coefficient.

The optimal cost-to-go, also called the value function, is defined as

$$J^\star = \min_{\boldsymbol{u}\in\mathcal{U}} J(\boldsymbol{u}), \tag{2.9}$$

which represents the minimum achievable cost starting from state $\boldsymbol{x}(t)$ at time t over the horizon $[t, T]$. Equivalently, the entire optimal control problem can be formulated by introducing a reward function $r(\boldsymbol{x}, \boldsymbol{u})$ and considering maximisation rather than minimisation. The formulation is fully general under the assumption $c = -r$.

Depending on the assumptions made in solving the minimisation problem, a wide range of methods arises. Examples of fundamental books discussing the developments of Optimal Control theory are Kirk (2004), Macki & Strauss (1982) and Berkovitz (2013).

2.3.1. *Hamilton-Jacobi-Bellman equation*

A central result in Optimal Control theory arises from the Dynamic Programming theory developed by R. Bellman in the 1950s (Bellman, 1954*a*). Within this framework, the optimal cost function $J^\star$ satisfies a nonlinear partial differential equation known as the Hamilton-Jacobi-Bellman (HJB) equation. The HJB equation provides necessary and sufficient conditions for optimality under appropriate regularity assumptions, and its solution corresponds to the optimal cost function. The HJB is given by:

$$-\frac{\partial J^\star}{\partial t} = \min_{\boldsymbol{u}\in\mathcal{U}} \Big\{ c(\boldsymbol{x}, \boldsymbol{u}) + \left(\nabla_{\boldsymbol{x}} J^\star\right)^\top f(\boldsymbol{x}, \boldsymbol{u}) \Big\}. \tag{2.10}$$

where $\frac{\partial}{\partial t}$ denotes the partial derivative with respect to time t and $\nabla_{\boldsymbol{x}} J^\star \in \mathbb{R}^{n_x}$ is the gradient of the value function with respect to the state vector $\boldsymbol{x}$. The plant dynamics f is here considered without external disturbances. Note that the discrete-time counterpart of the HJB equation is generally known as the Bellman optimality equation. This formulation formalises Bellman principle of optimality (Bellman, 1957), which states that the optimal cost-to-go from any initial state can be computed recursively from the optimal cost-to-go of the subsequent states. Exact solutions of the HJB exist under simplifying assumptions about the control problem, for instance, linear dynamics with quadratic cost functions, as discussed in the following section.

2.3.2. *Linear-quadratic methods*

The plant dynamics have been so far considered in their most general nonlinear form, as presented in Equation (2.1) for a continuous-time formulation. A special case arises when the dynamics are linear, or when linearised around an equilibrium (or stationary) point $\bar{\boldsymbol{x}}$, yielding a linear time-invariant representation:

$$\dot{\boldsymbol{x}} = \mathbf{A}\boldsymbol{x} + \mathbf{B}\boldsymbol{u}, \tag{2.11}$$

where $\mathbf{A} \in \mathbb{R}^{n_x \times n_x}$ and $\mathbf{B} \in \mathbb{R}^{n_x \times n_u}$ are called state and input matrices, respectively. These matrices are obtained through a first-order Taylor expansion of the nonlinear dynamics f in Equation (2.1) around the equilibrium point. Specifically,

$$\mathbf{A} = \left.\frac{\partial f}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\bar{\boldsymbol{x}},\, \boldsymbol{u}=\bar{\boldsymbol{u}}}, \quad \mathbf{B} = \left.\frac{\partial f}{\partial \boldsymbol{u}}\right|_{\boldsymbol{x}=\bar{\boldsymbol{x}},\, \boldsymbol{u}=\bar{\boldsymbol{u}}} \tag{2.12}$$

where $\bar{\boldsymbol{u}}$ is the input corresponding to the equilibrium state. A similar expression, even if not formally stated, can be extended to the discrete-time formulation, as done in the general nonlinear case in Equation (2.2). The linear formulation is introduced because a linear approximation of the dynamics has proven effective for control design even in moderately nonlinear systems. Examples include the application of linear control methods to the cylinder wake (Roussopoulos, 1993; Protas, 2004), boundary layers (Bagheri *et al.*, 2009*a*), cavity flows (Rowley & Williams, 2006), and channel flow (Bewley & Liu, 1998). Linear control methods in fluid dynamics have also been extensively reviewed, particularly concerning their applications in fluid mechanics (Bagheri *et al.*, 2009*b*; Kim & Bewley, 2007; Sipp *et al.*, 2010). The success of linear methods, even in moderately nonlinear cases linearised around a fixed point or a periodic orbit, can be explained as follows. The linear solution is generally valid in the neighbourhood of the point used for linearisation. Moreover, the majority of control applications target stabilising controllers, whose goal is to regulate the system to a region where the linear approximation becomes increasingly accurate.

The most considered strategy using the linear assumption is the Linear Quadratic Regulator (LQR). It is the solution of the HJB when the plant model is linear, the cost-to-go is expressed as a quadratic function of the states and input and the scenario of full state observability, meaning that the feedback to the state is characterised by the full state, so $\boldsymbol{y} = \boldsymbol{x}$ is considered. The quadratic cost function is defined as

$$J(\boldsymbol{u}) = \int_t^T \left[\boldsymbol{x}(\tau)^\top \mathbf{R}_x \boldsymbol{x}(\tau) + \boldsymbol{u}(\tau)^\top \mathbf{R}_u \boldsymbol{u}(\tau)\right] \,\mathrm{d}\tau + \boldsymbol{x}(T)^\top \mathbf{R}_T \boldsymbol{x}(T), \tag{2.13}$$

where $\mathbf{R}_x \in \mathbb{R}^{n_x \times n_x}$ and $\mathbf{R}_T \in \mathbb{R}^{n_x \times n_x}$ are symmetric positive semi-definite matrices, and $\mathbf{R}_u \in \mathbb{R}^{n_u \times n_u}$ is symmetric positive definite. These matrices respectively weight the state deviation over the horizon, the final state deviation at the terminal time, and the control effort. The solution of the HJB in Equation (2.10) leads to the optimal determination of the state-feedback control law, which is linear in the state and defined as

$$\boldsymbol{u} = \mathbf{K}\boldsymbol{x}, \tag{2.14}$$

$\mathbf{K} \in \mathbb{R}^{n_u \times n_x}$ is the optimal state-feedback gain, corresponding to a linear extension of the feedbak law introduced in Equation (2.4). In the LQR formulation, it is based on the solution of the Riccati equation (Anderson & Moore, 1990). However, the computational cost of solving the equation to get the feedback gain scales with the cube of the dimension of the full state n_x (Duriez *et al.*, 2017), making the problem challenging for high-dimensional systems.

In practical scenarios, the full system state as feedback is often not accessible, and only a subset of outputs or sensor measurements is available. This introduces an estimation problem. In the linear case, the *separation principle* applies, which states that the design of the optimal controller can be separated from the design of the optimal state estimator. Consequently, the optimal linear controller can be implemented using estimates of the state without compromising optimality.

Incorporating an optimal state estimator into the LQR problem leads to the formulation of the LQG controller, where the optimal linear estimator is the Kalman Filter (KF). The LQG thus combines the linear quadratic regulator with the KF to provide control based on estimated states rather than directly measured ones.

A detailed discussion of the KF derivation is beyond the scope of this dissertation. In **Paper 2**, an estimation problem is addressed within a model predictive control framework, but using a nonlinear Kalman estimator. Specifically, the Unscented Kalman Filter (UKF) has been used.

In the general framework of system-norm-based control, the LQG can be seen as an H_2-type controller, i.e. a controller aiming to minimise the average effect of stochastic process disturbances. The optimisation problem in this framework can be adjusted to promote robustness of the controllers against model uncertainty, and this leads to the formulation of the class of H_∞ controllers (Doyle *et al.*, 1988).

2.3.3. *Approximated Dynamic Programming solutions*

When considering general nonlinear systems, the continuous-time HJB equation is rarely solvable analytically, as the solution for high-dimensional dynamics quickly becomes intractable. A vast literature addresses approximate and practical solution methods, and the interested reader is referred to the comprehensive treatments in classical texts (Bellman, 1954*b*; Bertsekas, 2012; Powell, 2007). For brevity, only some relevant results are considered here.

Approximate solutions of the Bellman optimality equation naturally exploit its fixed-point property. Classical algorithms, such as Value Iteration (Bellman, 1957) and Policy Iteration (Howard, 1960), iteratively refine estimates of the optimal cost-to-go by successive approximation. Model-free and value-based RL approaches approximate the Bellman solution using data collected from system interactions (Semeraro, 2025), whereas model-based variants use a known or learned system model to propagate the Bellman recursion efficiently. Algorithms such as fitted value iteration, Dyna and many others instantiate this approach, providing a broad and versatile toolkit for control-oriented learning (Moerland *et al.*, 2023; Polydoros & Nalpantidis, 2017; Luo *et al.*, 2024*a*). Frameworks of RL, in both model-free and model-based formulations, have demonstrated considerable potential for control in fluid dynamics. A comprehensive overview of recent advances in RL applications within this domain is provided by Vignon *et al.* (2023). Significant contributions have been made in the context of two-dimensional wake flow control (Rabault *et al.*, 2019; Garcia *et al.*, 2025) and three-dimensional configurations (Chatzimanolakis *et al.*, 2024; Montalà *et al.*, 2025; Suárez *et al.*, 2025). Several studies have further investigated experimental RL for controlling the flow past bluff bodies and backwards-facing step (Fan *et al.*, 2020; Amico *et al.*, 2022; Castellanos *et al.*, 2022; Xiang *et al.*, 2024). Notable applications of model-based RL in flow control are described by the works by Dong *et al.* (2023), Ye & Elsheikh (2025) and Weiner & Geise (2025).

For control applications in which feedback consists of sparse and noisy measurements, RL often exhibits poorer performance, and the control task becomes more challenging. The literature reports several approaches that exploit temporal information to address the problem of Partially Observable Markov Decision Processes (Kurniawati, 2022). Examples include the works of Xia *et al.* (2024) and Wang *et al.* (2024). Other methods adopt alternative strategies, such as integrating Data Assimilation (DA, Ozan *et al.*, 2025), Long Short-Term Memory (LSTM) networks (Hausknecht & Stone, 2015), or attention-based mechanisms (Weissenbacher *et al.*, 2025) in the RL pipeline.

Model Predictive Control is closely related to RL (Bertsekas, 2005). Indeed, it performs finite-horizon optimisation using the system model at each control step. Conceptually, MPC can be interpreted as a rollout-based, value-function approximation: by solving a truncated optimisation problem over a receding horizon, it generates control actions that approximate the Bellman-optimal policy locally. A comprehensive treatment of MPC is provided in Chapter 3.

2.4. Challenges of turbulent flow models

The selection of an accurate model of the system dynamics is crucial for the design of a model-based controller. In fluid-dynamic applications, this requirement poses significant challenges due to the intrinsic complexity of turbulent flows, particularly in configurations with high Reynolds numbers.

The main difficulties in modelling turbulent flows arise from the high dimensionality of the state, strong nonlinearities, potential time delays, and multiscale phenomena in both space and time. The high dimensionality stands in the number of degrees of freedom that are necessary to accurately describe a fluid dynamics problem. High-fidelity numerical simulations of turbulent flows often require a spatial discretisation comprising tens of millions of points (Duriez *et al.*, 2017), rendering standard identification and model-based control methods computationally intractable. Nonlinearities in the dynamics further complicate the response of the system to actuation. Time delays introduce non-immediate responses to control inputs, and multiscale phenomena are governed by energy transfer mechanisms, such as the Kolmogorov cascade (Pope, 2000), where energy is transferred from coherent structures at large scales to progressively smaller scales until dissipation occurs. In some flows, an inverse cascade may also arise, in which smaller-scale structures merge to form larger, lower-frequency structures.

The model choice has profound implications for control design, as it requires carefully balancing competing factors such as accuracy, computational cost, spatial-temporal resolution, generality across operating regimes, and the effort required to construct or identify the model. Regardless of the identification approach, models can generally be categorised according to their fidelity and computational cost. A widely cited classification of models used in turbulence control is presented in Brunton & Noack (2015). In this work, the models are grouped into three main categories

- **White-box** models provide a detailed representation of all relevant flow features. They are typically very high-dimensional, often exceeding available memory and computational resources, which limits their applicability for real-time control and estimation. Numerical solution of the Navier-Stokes equations is typically performed within the framework of computational fluid dynamics. Common numerical methods include spectral elements, finite differences, finite volumes, and point-vortex methods, among others. For flows with high Reynolds numbers or complex geometries, Direct Numerical Simulation (DNS) becomes computationally prohibitive, necessitating modelling of smaller turbulent scales. This gives rise to Reynolds-averaged Navier-Stokes (Launder & Spalding, 1983) and large eddy simulation (Smagorinsky, 1963) approaches. These methods are excluded from consideration here, as their computational cost renders them unsuitable for deployment in real-time estimation and control.

- **Grey-box** models include a class of methodologies that approximate the flow using a reduced set of relevant coordinates, such as mode amplitudes or coherent structures, capturing the dominant dynamics while reducing computational cost. Common techniques to extract these modes include Proper Orthogonal Decomposition (POD, Berkooz *et al.*, 1993), Balanced POD (Rowley, 2005), and Dynamic Mode Decomposition (Schmid, 2010). A notable example is the Galerkin projection (Fletcher, 1984) of the Navier-Stokes equations onto a modal basis, which is frequently used to obtain nonlinear reduced-order models, such as POD-Galerkin models (Holmes *et al.*, 2012). These reduced-order models enable fast simulation and control, but their validity is typically limited to the dominant frequencies or known nonlinear mechanisms captured by the selected modes. More recently, grey-box modelling has been extended by approaches that integrate physical knowledge with data-driven learning. Sparse Identification of Nonlinear Dynamics (SINDy, Brunton *et al.*, 2016*a*) identifies governing equations from data using a sparsity hypothesis. Operator inference (Kramer *et al.*, 2024) learns reduced operators directly from data within a POD-Galerkin framework. Physics-informed neural networks (Raissi *et al.*, 2019) embed known physical laws, such as the Navier-Stokes equations, into neural network training.

- **Black-box** models are constructed purely from input-output data, without explicit knowledge of the underlying fluid dynamics. They are advantageous for rapid deployment, although they are generally characterised by limited interpretability. Among the techniques commonly employed in this category are the Eigensystem Realisation Algorithm (Juang & Pappa, 1985) and Observer Kalman Filter Identification (Juang *et al.*, 1993) using time-series data to identify input-output relations, Volterra series (Boyd *et al.*, 1984) for nonlinear mapping, and machine learning approaches such as Support Vector Machines (Schölkopf & Smola, 2002). Traditional parametric input-output models, such as autoregressive models (Box *et al.*, 2015; Billings, 2013), or output-error formulations (Ljung, 2003) also fall in this category.

The choice between white-, grey-, and black-box models represents a trade-off between computational cost and fidelity. The present dissertation focuses primarily on grey-box models, which inherently benefit from dimensionality reduction, and a discussion of the most relevant methods in this category is presented in the following section.

2.4.1. *Dimensionality reduction*

The challenges associated with modelling strategies in fluid flows motivate the adoption of identification techniques that operate on a reduced set of coordinates, giving rise to ROM approaches that balance computational efficiency and interpretability. Dimensionality reduction seeks to represent the essential dynamics on a lower-dimensional subspace or manifold, retaining the dominant flow features while alleviating the computational burden. This reduction is often motivated by the presence of coherent structures, which are connected regions of the flow where key quantities such as velocity or vorticity exhibit significant spatial and temporal correlation (Hussain, 1986; Robinson, 1991; Holmes *et al.*, 2012). A complementary viewpoint is offered by the manifold hypothesis, which assumes that high-dimensional flow data evolve on, or in proximity to, a low-dimensional manifold embedded within the state space (Belkin & Niyogi, 2001; Lin *et al.*, 2015).

Two similar approaches for dimensionality reduction are considered here. POD identifies the most energetic modes from a set of snapshots, producing a linear basis that captures coherent structures efficiently. By contrast, Isometric Mapping (ISOMAP) is a nonlinear, data-driven technique that seeks a low-dimensional embedding while preserving the intrinsic geometric relationships of the high-dimensional data. Being data-driven, both methods start from a dataset that captures the system dynamics. In the context of controlled flows, this dataset typically consists of state snapshots and corresponding control inputs collected either from experiments or high-fidelity simulations. The collection of M snapshots of the state vector and control inputs is arranged as

$$\mathbf{X} = \left[\boldsymbol{x}_1, \dots, \boldsymbol{x}_M\right] \in \mathbb{R}^{n_x \times M}, \quad \mathbf{U} = \left[\boldsymbol{u}_1, \dots, \boldsymbol{u}_M\right] \in \mathbb{R}^{n_u \times M}, \tag{2.15}$$

with $\mathbf{X}$ and $\mathbf{U}$ storing in columns the full states and the corresponding inputs related to one or more trajectories of the plant dynamics. Implicit in this construction is the assumption that the plant output coincides with the full state vector, i.e. that all state components are measured, or otherwise available, at each snapshot time.

2.4.1.1. *Proper Orthogonal Decomposition*

Proper Orthogonal Decomposition was originally developed in the context of multivariate statistics as Principal Component Analysis (Pearson, 1901; Hotelling, 1992) and later in fluid mechanics and dynamical systems to identify coherent structures (Lumley, 1967; Berkooz *et al.*, 1993). The fundamental principle of POD is to decompose a dataset into orthogonal modes, ordered according to the variance they capture. It seeks a linear subspace which is optimal for variance content. This decomposition allows the identification of the most significant structures in the system while discarding less relevant contributions. By representing the system in a lower-dimensional space spanned by these orthogonal modes, POD enables a tractable description of high-dimensional dynamics, preserving the information most relevant for analysis and control design.

POD is typically computed using Singular Value Decomposition (Wall *et al.*, 2003):

$$\mathbf{X} = \boldsymbol{\Phi}\, \boldsymbol{\Sigma}\, \boldsymbol{\Psi}^\top, \tag{2.16}$$

where $\boldsymbol{\Phi} \in \mathbb{R}^{n_x \times n_x}$ contains the spatial modes, $\boldsymbol{\Sigma} \in \mathbb{R}^{n_x \times M}$ is a diagonal matrix of singular values quantifying the variance captured by each mode, and $\boldsymbol{\Psi} \in \mathbb{R}^{M \times M}$ contains the temporal coefficients.

The optimal low-rank approximation of the data is obtained by truncating the decomposition to the first r modes

$$\mathbf{X}_r = \boldsymbol{\Phi}_r\, \boldsymbol{\Sigma}_r\, \boldsymbol{\Psi}_r^\top = \boldsymbol{\Phi}_r \mathbf{Q}, \tag{2.17}$$

with $\boldsymbol{\Phi}_r \in \mathbb{R}^{n_x \times r}$, $\boldsymbol{\Sigma}_r \in \mathbb{R}^{r \times r}$, and $\boldsymbol{\Psi}_r \in \mathbb{R}^{M \times r}$. The matrix $\mathbf{Q} = \boldsymbol{\Sigma}_r \boldsymbol{\Psi}_r^\top \in \mathbb{R}^{r \times M}$ represents the modal amplitudes, obtained by weighting each right singular vector by its corresponding singular value. This truncated decomposition is equivalent to solving the minimisation problem

$$\mathbf{X}_r = \arg \min_{\mathrm{rank}(\mathbf{Y}) = r} \|\mathbf{X} - \mathbf{Y}\|_F, \tag{2.18}$$

with $\|\cdot\|_F$ denoting the Frobenius norm and $\mathrm{rank}(\mathbf{Y})$ denoting the rank of the matrix $\mathbf{Y}$. A standard strategy for selecting the dimension of the latent space r, is to examine the decay of the singular values. In particular, the retained variance of the first r modes is quantified as the normalised cumulative sum of the squared singular values, denoted by σ_i:

$$\gamma_r = \frac{\sum_{i=1}^{r} \sigma_i^2}{\sum_{i=1}^{\min(n_x, M)} \sigma_i^2}. \tag{2.19}$$

The latent dimension r is then chosen to ensure that this retained variance exceeds a specified threshold, thereby capturing the bulk of the variance content (Brindise & Vlachos, 2017; Kramer *et al.*, 2024).

The truncated left singular vectors $\mathbf{\Phi}_r$, define a reduced basis that can be used to project the high-dimensional dynamics onto a lower-dimensional latent space. Each state vector $\boldsymbol{x}_k$ is then represented in this reduced basis as

$$\boldsymbol{q}_k = \mathbf{\Phi}_r^\top \boldsymbol{x}_k, \tag{2.20}$$

where $\boldsymbol{q}_k \in \mathbb{R}^r$ is the corresponding low-dimensional representation of $\boldsymbol{x}_k$. This formulation relies on the fact that $\mathbf{\Phi}_r$ forms an orthonormal basis.

2.4.1.2. *Isometric Mapping*

Isometric Mapping was introduced by Tenenbaum *et al.* (2000) and can be regarded as a nonlinear extension of the classical metric Multidimensional Scaling (MDS, Tzeng *et al.*, 2008). The main idea behind ISOMAP is to preserve the intrinsic geometry of high-dimensional data by maintaining geodesic distances along the underlying manifold rather than Euclidean distances. However, it is a discrete isometric embedding method, that is, it does not explicitly estimate the differential structure of the manifold nor provide guarantees for topological preservation or embedding smoothness. Since its introduction, ISOMAP has been widely applied across many fields, including computer vision (Souvenir & Pless, 2005), dynamical systems (Suetani *et al.*, 2012), and fluid mechanics (Tauro *et al.*, 2014; Farzamnik *et al.*, 2023; Amico *et al.*, 2025).

The method starts from a dataset of full states collected from the plant, as the one in Equation (2.15). The Euclidean distance matrix $\mathbf{D}_{\mathrm{E}} \in \mathbb{R}^{M\times M}$ is first computed, with entries corresponding to the distance between snapshots. In particular, the entry at the ith row and jth column is

$$\mathbf{D}_{\mathrm{E_{i,j}}} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2, \tag{2.21}$$

where $\|\cdot\|_2$ denotes the Euclidean norm.

The goal is to approximate the true geodesic distance matrix on the manifold in which the data lie. One method to approximate geodesic distances is to construct a neighbourhood graph using the k-Nearest Neighbours (kNN, Fix & Hodges, 1989) approach. Distances between nearby neighbours are approximated by Euclidean distances, and geodesic distances between non-neighbours are computed as the shortest path through neighbouring snapshots, for instance via Floyd's algorithm (Floyd, 1962).

The choice of the number of neighbours, denoted by k_e, is critical. If k_e is too low, the graph may become disconnected, preventing estimation of geodesic distances between all snapshots. Conversely, if k_e is too high, short-circuiting may occur, connecting snapshots with a direct route even if actually far apart on the manifold. The optimal k_e is typically chosen in a range between the minimum value ensuring connectivity and the maximum value avoiding short circuits. Sudden drops in the Frobenius norm of the geodesic distance matrix, which indicate the appearance of shortcuts in the graph, can be used to identify the upper bound.

Once the geodesic distance matrix, denoted by $\mathbf{D}_{\mathrm{G}}$, is estimated, MDS is applied. First, the double-centered matrix $\mathbf{B}$ is computed as

$$\mathbf{B} = -\frac{1}{2}\mathbf{H}(\mathbf{D}_{\mathrm{G}} \odot \mathbf{D}_{\mathrm{G}})\mathbf{H}, \tag{2.22}$$

where $\mathbf{H} = \mathbf{I}_M - \frac{1}{M}\mathbf{1}_M\mathbf{1}_M^T$ is the centering matrix, $\mathbf{I}_M$ the identity matrix, $\mathbf{1}_M$ a vector of ones, and $\odot$ denotes the Hadamard product. The eigendecomposition of $\mathbf{B}$ the yields

$$\mathbf{B} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T, \tag{2.23}$$

where $\mathbf{V}$ contains the eigenvectors and $\mathbf{\Lambda}$ the corresponding eigenvalues. Selecting the r largest positive eigenvalues $\mathbf{\Lambda}_r$ and their eigenvectors $\mathbf{V}_r$, the low-dimensional coordinates are stored in the matrix

$$\mathbf{\Gamma} = \mathbf{V}_r \mathbf{\Lambda}_r^{1/2}. \tag{2.24}$$

where $\mathbf{\Lambda}_r^{1/2}$ is the diagonal matrix of square roots of the retained eigenvalues. This embedding corresponds to minimising the cost function

$$\min_{\mathbf{\Gamma}} \|\mathbf{\Gamma}\mathbf{\Gamma}^T - \mathbf{B}\|_F^2. \tag{2.25}$$

Finally, the appropriate latent space dimensionality r can be determined using the residual variance (Tenenbaum *et al.*, 2000):

$$R_v = 1 - R^2(\mathrm{vec}(\mathbf{D}_\mathrm{G}), \mathrm{vec}(\mathbf{D}_\Gamma)), \tag{2.26}$$

where $R^2(\cdot,\cdot)$ denotes the squared correlation coefficient, $\mathrm{vec}(\cdot)$ the vectorisation operator, and $\mathbf{D}_\Gamma$ the Euclidean distance matrix between points in the low-dimensional embedding. Typically, r is chosen by identifying an elbow in the residual-variance curve, balancing dimensionality reduction and distance preservation. The residual variance indeed quantifies how well the low-dimensional embedding preserves the structure of the data, as encoded by the topology of the geodesic distance matrix.

In ISOMAP, as in many other manifold learning techniques, a decoder is not provided (Mendez, 2023). Nonetheless, several strategies exist to reconstruct the original high-dimensional states from the low-dimensional embedding. Common approaches include interpolation-based methods, such as kNN (Farzamnik *et al.*, 2023) or kernel regression techniques, as well as projection-based methods that exploit the eigenvectors from the MDS step (Bakır *et al.*, 2004). The choice of decoding strategy depends on the desired trade-off between computational efficiency and reconstruction accuracy, as well as on the density and coverage of the low-dimensional embedding. These reconstruction methods, however, do not guarantee that the reconstructed state preserves the physical properties or invariants of the system. To evaluate how closely the reconstructed state satisfies the physics, it is necessary to define a suitable error metric, such as the residual of the governing equations, if available. This a posteriori estimation quantifies the deviation of the reconstruction from the true system dynamics. A promising approach is to perform decoding as a pre-image problem (Kwok & Tsang, 2004; Arias *et al.*, 2007; Arif *et al.*, 2010), where it is formulated as an optimisation process. Such methods allow the incorporation of physics-based loss terms in the reconstruction, although they may scale poorly with the dimensionality of the application.

ISOMAP effectively captures low-dimensional, nonlinear structure when the manifold is relatively simple and well sampled (Mendez, 2023; Niu *et al.*, 2025). When compared to other nonlinear dimensionality reduction methods for pattern identification and data compression, ISOMAP demonstrates superior performance over many techniques, including linear approaches such as POD, Kernel Principal Component Analysis, Multi-Dimensional Scaling, and Local Linear Embedding (Mendez, 2023; Farzamnik *et al.*, 2023). Several reviews have surveyed methodologies in nonlinear manifold learning, generally concluding that the choice of algorithm should be guided by the specific requirements and constraints of the application (Zahed & Skafyan, 2025; Bartenhagen *et al.*, 2010). However, for multiscale or turbulent flows, it presents some limitations. Topological instability and difficulties with non-convex or highly curved manifolds can distort the embedding of interacting scales (Niu *et al.*, 2025). It is also characterised by poor extrapolation beyond the training neighbourhoods, which can lead to inaccurate reconstruction of fine-scale structures. Some alternative approaches for handling these challenges are discussed in Niu *et al.* (2025).

Beyond these topological complexities, the reliability of the manifold learned through ISOMAP depends on the data, which in practical experimental scenarios can be sparse or noisy. Standard ISOMAP estimates geodesic distances via Nearest-Neighbour graphs, making it particularly sensitive to such anomalies. In practice, these effects can lead to short-circuit errors in the neighbourhood graph or disconnected regions if the sampling is too sparse, reducing the ability of ISOMAP to accurately uncover the intrinsic nonlinear manifold underlying the data. To mitigate these limitations, several strategies have been proposed. Preprocessing the data, for instance, through POD for noise reduction, can improve the quality of the data. Careful selection of the neighbourhood size, including adaptive versions of the algorithm (Mekuz & Tsotsos, 2006), can also help maintain connectivity while respecting local manifold structure. In addition, more robust alternatives to standard ISOMAP have been developed to address sparse and irregular sampling. These include Parallel Transport Unfolding (Budninskiy *et al.*, 2019), geodesic distance estimation based on Geodesic Forests (Madhyastha *et al.*, 2020), Smooth Geodesic Embedding (Gajamannage *et al.*, 2019), Robust Kernel Isomap (Choi & Choi, 2007), and other extended implementations (Yousaf *et al.*, 2024).

2.4.2. *Autoencoders*

A recent class of nonlinear methods for reduced-order modelling involves autoencoders, which aim to compute a compressed representation of high-dimensional data while preserving as much information as possible (Mendez, 2023). When implemented with neural networks, these models are referred to as deep autoencoders, providing a flexible framework for discovering low-dimensional latent representations.

Conceptually, a deep autoencoder consists of two coupled parametric functions: an encoder $F_e(\boldsymbol{x}, \mu_e)$, parameterised by μ_e, which maps the full state $\boldsymbol{x} \in \mathbb{R}^{n_x}$ to its low-dimensional latent coordinate $\boldsymbol{q} \in \mathbb{R}^r$, and a decoder $F_d(\boldsymbol{q}, \mu_d)$, parameterised by μ_d, which reconstructs the high-dimensional state from the latent representation. The parameters of these functions are optimised simultaneously by minimising a reconstruction loss, typically the squared error between the input and output, so that

$$\boldsymbol{x} \approx F_d(F_e(\boldsymbol{x}, \mu_e), \mu_d), \quad \boldsymbol{q} = F_e(\boldsymbol{x}, \mu_e) \in \mathbb{R}^r. \tag{2.27}$$

The encoder can thus be interpreted as a parametric mapping from the full state to a low-dimensional manifold, while the decoder defines a parametric reconstruction operator that approximates the inverse mapping. This framework allows the identification of nonlinear embeddings that capture essential structures in complex datasets, offering capabilities for control, estimation and compression of fluid dynamics data.

Despite their expressive power, autoencoders generally require large amounts of training data, are computationally intensive, and often lack direct interpretability, since the latent coordinates do not necessarily correspond to physically meaningful modes. In this respect, it is worth noting that POD can be interpreted as a particular case of a linear autoencoder with a single hidden layer and linear activation functions, which yields an optimal low-dimensional representation in the least-squares sense Mendez (2023).

Recent studies have demonstrated the effectiveness of autoencoders for low-order representations in a range of fluid-dynamics applications. For flow compression and reconstruction, convolutional and physics-informed autoencoders have been employed for three-dimensional flow fields from sparse or high-dimensional data. These approaches aim to preserve the key flow features while achieving an efficient low-dimensional representation (Guo *et al.*, 2016; Fukami *et al.*, 2020; Uribarri & Mindlin, 2020; Fukami & Taira, 2023). In nonlinear reduced-order modelling, variational autoencoders and attention-augmented architectures have enabled compact latent representations that capture unsteady and turbulent dynamics more accurately than linear methods (Solera-Rico *et al.*, 2024; Beiki & Kamali, 2025). Furthermore, recent work has shown that the latent space can be interpreted in terms of intrinsic system features (Page *et al.*, 2021; Mo *et al.*, 2024), greatly enhancing the interpretability of these methods. For data-driven state estimation and control, autoencoder-based frameworks combined with predictive methods allow fast inference and temporal prediction of complex flow fields, supporting both analysis and control design. Some approaches include autoencoders coupled with Echo State Networks for temporal prediction (Racca *et al.*, 2023; Özalp *et al.*, 2026), Koopman-based linear models (Lusch *et al.*, 2018), LSTM networks (Nakamura *et al.*, 2021; Hasegawa *et al.*, 2020), Transformer architectures (Mentzelopoulos *et al.*, 2024), and Neural Ordinary Differential Equations (Linot & Graham, 2022), among others (Herzog *et al.*, 2019; Vlachas *et al.*, 2022). These results illustrate the growing versatility of autoencoder-based approaches in fluid mechanics.

2.4.3. *Data Assimilation*

As noted in the formulation of the control problem, the full state of a dynamical system is often not directly accessible through measurements. To overcome this limitation, DA provides a systematic framework to estimate the full system state by combining observations from the plant with information from a dynamical model of the system (Asch *et al.*, 2016).

Classical DA methods can be broadly classified into two main categories. Variational methods formulate the assimilation problem as the minimisation of a cost function, typically measuring the discrepancy between model predictions and observations over a time window. In contrast, sequential methods update the state estimate recursively as new observations become available. A comprehensive review of classical DA techniques is provided in Bannister (2017).

Recent advances in machine learning have significantly enhanced DA by improving computational efficiency and model expressiveness. Several reviews document the integration of machine learning and data assimilation (Buizza *et al.*, 2022; Cheng *et al.*, 2023). One notable approach within this framework is latent space data assimilation, where the high-dimensional system state is projected onto a low-dimensional manifold before assimilation. This is typically achieved using dimensionality reduction techniques such as POD or autoencoders, for instance, allowing assimilation to be performed at a reduced computational cost. Applications include variational DA or sequential approaches in latent space using POD (Maulik *et al.*, 2022; Heaney *et al.*, 2024; Nóvoa & Magri, 2025) or autoencoders (Amendola *et al.*, 2021; Peyron *et al.*, 2021; Özalp *et al.*, 2026). These methods are particularly relevant in the context of low-dimensional control, where they enable the design of controllers even in the presence of partial, sparse, or noisy measurements.

2.5. System identification with data-driven strategies

Once the system coordinates are known (either in full-state description or in latent space form), the model plant must be identified. Owing to the limitations of white modelling in fluid flows discussed in Section 2.4, in this section, two strategies for data-driven system identification will be presented.

The SINDy framework was first introduced by Brunton *et al.* (2016*a*) and subsequently extended to systems with control inputs (Brunton *et al.*, 2016*b*) as well as to noise-robust formulations (Fasel *et al.*, 2022). SINDy seeks parsimonious models by identifying the fewest active terms from a library of candidate functions. Since its introduction, it has been successfully applied to a wide range of problems, including learning strategies for time-dependent systems (Quade *et al.*, 2018), uncovering dynamics in latent space representations (Fukami *et al.*, 2021; Loiseau & Brunton, 2018), and supporting control design (Kaiser *et al.*, 2018).

In contrast, the OpInf framework (Kramer *et al.*, 2024), was originally proposed by Peherstorfer & Willcox (2016) as a data-driven methodology for high-dimensional systems, explicitly designed to learn reduced-order models by projecting the dynamics onto a low-dimensional latent space via POD. OpInf has been applied to a variety of problems, including non-intrusive reduced-order modelling of incompressible flows (Benner *et al.*, 2020*a*; Rocha *et al.*, 2023), systems with non-polynomial nonlinearities (Benner *et al.*, 2020*b*) or low-dimensional models from noisy data (Uy *et al.*, 2023).

Together, SINDy and OpInf illustrate complementary strategies for data-driven system identification. SINDy emphasises sparsity and interpretability in rich feature spaces, whereas OpInf exploits dimensionality reduction to efficiently learn predictive operators in low-dimensional latent spaces. Throughout this exposition, the conceptual similarities and key differences between these two methodologies will be explicitly discussed, providing a unified perspective on their roles in constructing reduced-order dynamical models suitable for control design.

The objective of these data-driven strategies is to infer a dynamical model of the plant by using the information stored in the matrices $\mathbf{X}$ and $\mathbf{U}$ presented in Equation (2.15). Before entering the discussion on the two methodologies, the matrix of the time derivatives of the states is introduced:

$$\dot{\mathbf{X}} = \begin{bmatrix} \dot{\boldsymbol{x}}_1, & \dots, & \dot{\boldsymbol{x}}_M \end{bmatrix} \in \mathbb{R}^{n_x \times M}, \tag{2.28}$$

where $\dot{\boldsymbol{x}}_k$ is the derivative of the state vector evaluated at time t_k. The time derivatives, if not measured directly, can be estimated by numerical differentiation or approximated, for instance, using total variation regularised derivatives (Kaiser *et al.*, 2018).

The general approach considered is the one of approximating the dependencies of the plant dynamics, identifying a feature mapping denoted by $\boldsymbol{\kappa}(\boldsymbol{x}, \boldsymbol{u})$ and defined by

$$\boldsymbol{\kappa}(\boldsymbol{x}, \boldsymbol{u}) = \begin{bmatrix} \kappa_1(\boldsymbol{x}, \boldsymbol{u}) \\ \vdots \\ \kappa_{n_f}(\boldsymbol{x}, \boldsymbol{u}) \end{bmatrix} \in \mathbb{R}^m, \tag{2.29}$$

where $\kappa_1, \dots, \kappa_{n_f}$ are general linear or nonlinear functions of the state and input. These may include polynomials, trigonometric functions, or other nonlinear basis functions of the state and input, which are useful for representing the dynamics.

A model can then be described compactly as

$$\dot{\hat{\boldsymbol{x}}} = \boldsymbol{\Theta}\, \boldsymbol{\kappa}(\hat{\boldsymbol{x}}, \boldsymbol{u}), \tag{2.30}$$

where $\boldsymbol{\Theta} \in \mathbb{R}^{n_x \times m}$ collects the coefficients of the different functional dependencies included in the feature mapping in Equation (2.29) to be identified from data. Here, the notation $\hat{\boldsymbol{x}}$ denotes the state estimated by the identified model based on the learned dynamics. OpInf and SINDy propose similar frameworks to solve the data-driven model identification problem, even though with fundamental differences, which are discussed here.

2.5.1. *Sparse Identification of Nonlinear Dynamics*

SINDy aims at discovering a parsimonious model of the system by selecting only a few relevant terms from the library of candidate functions included in Equation (2.29). In the context of the general formulation above, the feature mapping $\boldsymbol{\kappa}(\boldsymbol{x}, \boldsymbol{u})$ is evaluated on the collected dataset of snapshots included in $\mathbf{X}$ and $\mathbf{U}$ to form a feature matrix

$$\boldsymbol{\Xi} = \begin{bmatrix} \boldsymbol{\kappa}(\boldsymbol{x}_1, \boldsymbol{u}_1) & \dots & \boldsymbol{\kappa}(\boldsymbol{x}_M, \boldsymbol{u}_M) \end{bmatrix} \in \mathbb{R}^{m \times M}. \tag{2.31}$$

SINDy then formulates the identification problem as a sparse regression, seeking a coefficient matrix $\boldsymbol{\Theta} \in \mathbb{R}^{n_x \times m}$ that explains the evolution of the states while retaining only the most significant terms. In particular, it estimates the matrix of coefficients $\boldsymbol{\Theta}$ by solving the following optimisation problem

$$\hat{\boldsymbol{\Theta}} = \arg\min_{\boldsymbol{\Theta}} \|\dot{\mathbf{X}} - \boldsymbol{\Theta}\, \boldsymbol{\Xi}\|_2^2 + \lambda \|\boldsymbol{\Theta}\|_1, \tag{2.32}$$

where λ is the sparsity-promoting coefficient penalising the L_1 norm of $\boldsymbol{\Theta}$, encouraging many coefficients to be exactly zero.

After the optimisation problem, the estimated matrix of coefficients $\hat{\boldsymbol{\Theta}}$ is used in the Equation (2.30) to obtain the plant model. SINDy is particularly effective when the underlying dynamics are governed by a few active terms in the chosen feature space. Its sparsity-promoting approach naturally mitigates overfitting and increases the interpretability of the learned model. In the standard definition of the algorithm, the optimisation in Equation (2.32) is solved using the Least Absolute Shrinkage and Selection Operator method (Tibshirani, 1996) or the Sequentially Thresholded Least Squares Procedure (Brunton *et al.*, 2016*a*).

2.5.2. *Operator Inference*

OpInf first compresses the high-dimensional state using the reduced basis $\boldsymbol{\Phi}_r \in \mathbb{R}^{n_x \times r}$ obtained via POD as explained in Section 2.4.1.1, retaining the first r modes. The system dynamics are then learned directly in this low-dimensional latent space spanned by the leading POD modes.

The dataset of state snapshots $\mathbf{X}$ is then first used to obtain its low-dimensional counterpart via projection as in Equation (2.20). Each latent coordinate is then collected in the matrix $\mathbf{Q}$, defined as

$$\mathbf{Q} = [\boldsymbol{q}_1, \dots, \boldsymbol{q}_M] \in \mathbb{R}^{r \times M}. \tag{2.33}$$

Similarly, the time derivative of the latent coordinates is obtained via the projection of the matrix $\dot{\mathbf{X}}$ as $\dot{\mathbf{Q}} = \boldsymbol{\Phi}_r^\top \dot{\mathbf{X}}$.

The feature mapping in OpInf is evaluated not on the high-dimensional space dataset but on its low-dimensional counterpart $\mathbf{Q}$. The feature matrix here is defined as

$$\boldsymbol{\Lambda} = \begin{bmatrix} \boldsymbol{\kappa}(\boldsymbol{q}_1, \boldsymbol{u}_1) & \dots & \boldsymbol{\kappa}(\boldsymbol{q}_M, \boldsymbol{u}_M) \end{bmatrix} \in \mathbb{R}^{m \times M}. \tag{2.34}$$

The reduced-order dynamics are then expressed directly in the low-dimensional space spanned by the POD modes

$$\dot{\hat{\boldsymbol{q}}} = \mathbf{O}\, \boldsymbol{\kappa}(\hat{\boldsymbol{q}}, \boldsymbol{u}), \tag{2.35}$$

where $\mathbf{O} \in \mathbb{R}^{r \times m}$ is the operator matrix to be inferred and $\hat{\boldsymbol{q}}$ indicates the latent state estimated by the identified model. In practice, in OpInf $\boldsymbol{\kappa}$ is typically restricted to polynomial terms (linear, quadratic, cubic) and state-input interactions, although other feature functions could be included. This restriction is motivated by two main reasons: first, the use of lifting transformations to render non-polynomial systems polynomial, and second, the fact that many numerical models arising in fluid mechanics naturally have a polynomial structure (Kramer *et al.*, 2024).

Different from SINDy, the OpInf problem is formulated as the following regularised least-squares optimisation

$$\hat{\mathbf{O}} = \arg\min_{\mathbf{O}} \|\dot{\mathbf{Q}} - \mathbf{O}\,\boldsymbol{\Lambda}\|_F^2 + \lambda_o \|\mathbf{O}\|_F^2, \tag{2.36}$$

where λ_o penalises large operator coefficients to improve generalisation. Equation (2.36) considers a ridge regularisation even though other forms of regularisation, such as Tikhonov regularisation, can also be used to improve numerical stability, as discussed in Kramer *et al.* (2024). Once $\hat{\mathbf{O}}$ has been identified, the reduced-order model can be propagated in time using the Equation (2.35) and the full-state approximation is recovered through $\hat{\boldsymbol{x}} = \boldsymbol{\Phi}_r \hat{\boldsymbol{q}}$.

It is important to note that the discussion within this thesis on SINDy and OpInf was primarily focused on learning continuous-time models. Without loss of generality, the formulation can be expressed in discrete-time by replacing the derivative matrix in Equation (2.28) with the shifted snapshot matrix

$$\begin{bmatrix} \boldsymbol{x}_2, & \dots, & \boldsymbol{x}_{M+1} \end{bmatrix} \in \mathbb{R}^{n_x \times M}, \tag{2.37}$$

allowing the identification of a discrete-time map with an integration interval Δt as in Equation (2.2).

The choice between continuous- and discrete-time dynamics for system identification and control involves multiple considerations. A major advantage of continuous-time models lies in their interpretability, since the governing equations are directly learned. This provides an easier understanding of the system dynamics, allows the incorporation of physics-based constraints or regularisation during learning, and facilitates the enforcement of invariants such as energy conservation or symmetries. Furthermore, once the continuous-time model is learned, the time discretisation is not fixed in advance, allowing integration at arbitrary time steps and providing flexibility in selecting control or simulation intervals. Continuous-time models do, however, present some challenges. Estimating the derivatives of the measured data (Equation 2.28) can amplify noise, particularly when using finite difference approaches. Techniques such as Local Polynomial Regression or methods based on total-variation regularisation exist to infer derivatives while mitigating noise sensitivity (Fan & Gijbels, 1996; Chartrand, 2011), yet this remains a non-trivial aspect. Another limitation is the computational cost: reproducing system trajectories requires numerical integration, which can be particularly demanding for stiff dynamics that require small integration time steps.

Discrete-time models, in contrast, are black-box mappings that predict the state after a fixed sampling interval. They are generally simpler to train and more robust to measurement noise, since no differentiation is required. However, they lack direct physical interpretability and are tied to the sampling interval used during identification. This dependency reduces flexibility when the control timestep changes during operation or needs adjustment. Long-horizon predictions can also accumulate errors more rapidly, especially for systems with large sampling intervals. In this thesis, the first contribution adopts a continuous-time formulation to take advantage of the interpretability of the dynamics, particularly in the presence of prior knowledge on the control of the system. The second contribution relies on a discrete-time model. Ultimately, the choice between continuous- and discrete-time approaches reflects a trade-off dictated by the specific constraints and objectives of the application.

SINDy and OpInf are used successfully for the test cases analysed in this thesis. However, some clarifications are required regarding their scalability to more complex applications, such as spatio-temporally chaotic and multiscale systems. In such problems, the effective number of dynamically interacting degrees of freedom is very large, and the dimensionality of the regression problem underlying these methods increases rapidly with system size. As a consequence, the candidate function libraries required for system identification can become prohibitively large, making direct application of sparse regression ill-posed or computationally intractable. Moreover, simple polynomial libraries may not be sufficiently expressive to capture the full range of interactions present in multiscale turbulent flows.

Nevertheless, these methods may remain valid in more complex settings for different, more limited reasons. In particular, it is often possible to identify possibly reduced coordinates that are suitable for control purposes, in which the dynamics can be simplified or expressed in a specific functional form. This is the case, for example, when identifying coordinates aligned with large-scale coherent structures of the flow. The dynamics of such reduced representations may admit sparse or low-order polynomial models that are amenable to SINDy or OpInf. Examples include the application of OpInf to large-scale combustor simulations (Kramer *et al.*, 2024), as well as the use of SINDy in combination with POD (Loiseau & Brunton, 2018) or nonlinear autoencoder-based dimensionality reduction (Fukami *et al.*, 2021; Bakarji *et al.*, 2023; Gao & Kutz, 2024).

In addition, models that are useful for control are not necessarily optimal in the sense of full-state estimation or long-time prediction. As famously stated by Box (1979), "*all models are wrong, but some are useful*". This highlights that, in control-oriented modelling, usefulness is determined by the ability of the model to capture the mechanisms that are most relevant for actuation and feedback, which are inherently application-dependent. This perspective is consistent with the modelling approach adopted in this thesis, where MPC is employed as the primary model-based control strategy. MPC can be interpreted as a numerical approximation to the solution of the HJB equation as stated in Section 2.3.3, in which the value function is approximated through finite-horizon rollouts. Consequently, the underlying dynamical model is only required to be accurate over a relatively short prediction horizon and in the neighbourhood of the current state, which is continuously observed or corrected through feedback. In this regime, local approximations of the dynamics can be sufficient and computationally effective, even if they do not constitute globally accurate models of the full system dynamics.

Chapter 3

Model Predictive Control

This chapter introduces Model Predictive Control, a model-based control strategy that has become a cornerstone of modern control engineering. Building upon the theory discussed in Chapter 2, in particular the foundations of optimal control, this chapter reviews the basic principles of MPC and presents its formulation as a receding-horizon optimal control problem. Special attention is given to MPC applications in fluid flows, highlighting both the practical advantages of the approach and the challenges that arise when dealing with high-dimensional and nonlinear dynamics. The chapter further discusses the limitations of standard implementations and outlines more advanced developments aimed at enabling real-time control through learning-based strategies. Finally, the conceptual connections between MPC and DP solutions like RL, especially through the lens of the Bellman equations, are examined.

3.1. What is Model Predictive Control?

The term Model Predictive Control is also known as Receding Horizon Control. It refers to a general class of model-based controllers that use a plant model to determine control actions by optimising over a recurrently shifting horizon, from which the term *receding-horizon* control derives. The origins of this approach can be traced back to early studies in the 1960s, such as the work by Propoi (1963), which applied linear programming methods to the synthesis of sampled-data systems. In this work, the formulation of a control problem as a moving-horizon controller was first introduced. Only one decade later, it was starting to be applied in practical scenarios. The concept of Model Predictive Heuristic Control was introduced by Richalet *et al.* (1978), formalising the idea of using a moving finite-horizon optimisation for the control of industrial processes. This methodology evolved a few years later with the development of Dynamic Matrix Control (Cutler & Ramaker, 1980), based on a linear step-response model that predicts future outputs from past inputs. This method was primarily targeted at the oil refining industry. The inclusion of quadratic programming in this framework was later introduced by Garcia & Morshedi (1986), leading to Quadratic Programming Dynamic Matrix Control, which made MPC practically applicable to multivariable chemical processes. Subsequently, robust formulations were proposed to establish theoretical studies for MPC application in industry (Campo & Morari, 1987; Bemporad & Morari, 2007). These addressed mainly the problems arising from plant model uncertainties and disturbances in the control process.

Over the past decades, MPC has found widespread applications across diverse domains, including power electronics (Vazquez *et al.*, 2014), building climate control (Oldewurtel *et al.*, 2012), agriculture (Ding *et al.*, 2018), mechatronic systems (Shi & Zhang, 2021), and automotive engineering (Hrovat *et al.*, 2012).

This extensive applicability reflects MPC's flexibility and its ability to control a wide variety of plant systems, including both linear and nonlinear dynamics. However, the primary feature driving its widespread adoption is its ability to explicitly incorporate constraints into the control formulation. In recent years, MPC has been increasingly applied to safety-critical systems, where violations of constraints or system failures can have severe consequences. This explains its success in fields such as robotic systems operating in uncertain environments (Zeng *et al.*, 2021).

3.2. The control strategy

In this section, the operational principles of MPC are presented, along with practical implementation considerations. A discussion of the main advantages and limitations of the approach is also provided.

3.2.1. *Receding Horizon control*

Falling under the category of model-based strategies, MPC computes control actions by solving, at each time step, a finite-horizon optimal control problem. Rather than synthesising a static feedback law, MPC explicitly predicts the future system behaviour over a horizon and optimises a sequence of future control inputs. Only the first control action of the optimal sequence is applied, and the optimisation is repeated at the next sampling time following a so-called receding-horizon strategy.

The main steps implemented by MPC consist on a sequence of operations executed at each discrete control instant t_k, as illustrated in Figure 3.1, and described as follows:

1. **State measurement**. The current system state $\boldsymbol{x}_k$ is either directly observed or estimated from available sensors. This measurement provides the initial condition for the prediction of future system behaviour, which is denoted by $\hat{\boldsymbol{x}}_k$.
2. w_p**-steps lookahead**. Using the measured or estimated state $\hat{\boldsymbol{x}}_k$ and a plant model, the future evolution of the system over a finite *prediction horizon* of w_p steps is forecasted. The prediction horizon is represented by the central grey region in Figure 3.1. The prediction of a plant trajectory starting from the specified initial condition using the model is generally referred to as *rollout*. This yields a sequence of states denoted by $\hat{\boldsymbol{x}}_{i+k}$ with $i = 1, \ldots, w_p - 1$. This prediction accounts for candidate control actions, typically defined over a control horizon $w_c \leq w_p$. These are denoted by $\hat{\boldsymbol{u}}_{i+k}$ with $i = 1, \ldots, w_c - 1$. In the case $w_c < w_p$, the final control inputs are held constant for the remaining steps of the prediction horizon, i.e. $\hat{\boldsymbol{u}}_{i+k} = \hat{\boldsymbol{u}}_{w_c-1+k}$, $i = w_c, \ldots, w_p - 1$.
3. **Cost function definition**. The information coming from a model rollout is used to build a cost function to evaluate the quality of candidate control sequences. This is defined as

$$J(\hat{\boldsymbol{u}}_k, \ldots, \hat{\boldsymbol{u}}_{k+w_c-1}; \hat{\boldsymbol{x}}_k) = \sum_{i=0}^{w_p-1} c(\hat{\boldsymbol{x}}_{k+i}, \hat{\boldsymbol{u}}_{k+i}) + h(\hat{\boldsymbol{x}}_{k+w_p}), \tag{3.1}$$

   where $c(\boldsymbol{x}, \boldsymbol{u})$ is the cost penalising deviations from desired trajectories and control effort at each step, and $h(\boldsymbol{x})$ represents a terminal cost that approximates performance beyond the prediction horizon (shaded blue region in Figure 3.1).

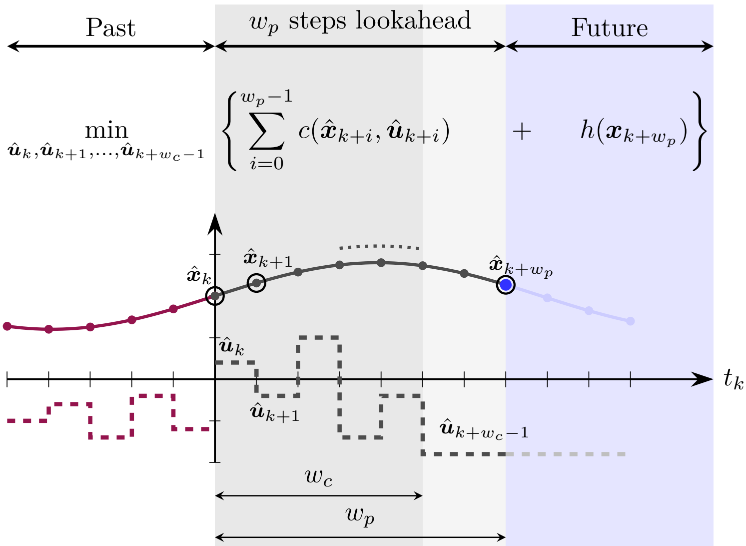

Figure 3.1: Receding-horizon strategy in MPC: starting from the current state $\hat{\boldsymbol{x}}_k$, directly measured or estimated, a w_p-step prediction of states is obtained by integrating the plant model under a candidate control sequence. The cost is the sum of stage costs along this predicted trajectory plus a terminal penalty at $\hat{\boldsymbol{x}}_{k+w_p}$. After optimisation, only the first control input is applied, and the optimisation is repeated at the next sampling instant.

4. **Optimization of control inputs**. A sequence of optimal control actions, $\boldsymbol{u}_k^{opt}, \dots, \boldsymbol{u}_{k+w_c-1}^{opt}$, is computed from the minimisation of the predefined cost function in Equation (3.1). In performing the optimisation, constraints on states and inputs, such as physical limits, maximum rates of change, or energy constraints, can be easily incorporated. Formally, the optimisation can be done by constraining the states and the inputs to the subsets $\mathcal{X} \subseteq \mathbb{R}^{n_x}$ for the states and $\mathcal{U} \subseteq \mathbb{R}^{n_u}$ for the inputs, defined depending on the specific application.
5. **Control implementation**. From the sequence of optimal control actions obtained, only the first term, $\boldsymbol{u}_k^{opt}$, is applied to the system. The rest of the action is discarded.
6. **Receding horizon update**. The system state is re-measured (or re-estimated from feedback), and the prediction-optimisation-control cycle is repeated at the next time step t_{k+1} (starting again from point 1).

3.2.2. *Practical considerations in Model Predictive Control*

Having introduced the basic formulation of MPC, several practical aspects related to its implementation must be addressed. The overall structure of MPC can be interpreted as an *offline learning and online play strategy*, following the terminology of Bertsekas (2024). In this interpretation, the offline phase refers primarily to the identification of the prediction model either from physics principles or data and, when applicable, to the definition of terminal cost functions or consideration regarding the stability of the closed-loop controllers. The online phase corresponds to the repeated approximation of the finite-horizon cost and its optimisation at each sampling instant.

A central component of MPC is the terminal cost function $h(\boldsymbol{x})$, in Equation (3.1), which accounts for the portion of the infinite-horizon objective beyond the prediction window. A correctly selected terminal cost contributes to approximating the infinite-horizon problem with the finite-horizon approach (Mayne *et al.*, 2000) solved by MPC or to guarantee closed-loop stability of the controller. In general, when the prediction horizon w_p is sufficiently long, the terminal cost expression can often be relatively simple. One common strategy is to choose a quadratic approximation derived by linearising the dynamics around an equilibrium and solving the associated Riccati equation (Rawlings *et al.*, 2020), as detailed in Section 2.3.2 for the LQR framework. More recently, learning-based methods have been proposed to better approximate terminal costs offline through dynamic programming or machine learning, enabling shorter prediction horizons (Abdufattokhov *et al.*, 2021; Moreno-Mora *et al.*, 2023).

Stability and robustness are fundamental considerations in practical MPC design. Closed-loop stability requires that the resulting state trajectories remain bounded for all future times under admissible controls. Although infinite-horizon formulations offer a conceptually straightforward route to stability, they are not practical. Finite-horizon stability is therefore enforced through additional constraints or penalty terms referred to as *stability constraints*. A detailed discussion on stability in the closed loop can be found in Findeisen *et al.* (2003). Robustness concerns instead arise from model mismatch and from disturbances such as measurement noise, unmodelled dynamics, or parameter uncertainties. It is useful to distinguish two notions. The first is the *inherent robustness* of nominal MPC implementation, arising from feedback and re-optimisation, while the other is the *explicit robustness* incorporated directly in the control design. Strategies such as min-max MPC and tube-based MPC (Bemporad & Morari, 2007; Saltık *et al.*, 2018) provide systematic ways to guarantee constraint satisfaction and stability under uncertainty. These methods, however, typically increase computational complexity, requiring a trade-off between robustness and real-time feasibility.

Another important aspect in the implementation of MPC is the need for the system state at every control timestep. When only output measurements are available, *output-feedback* MPC is employed. In this case, the state $\hat{\boldsymbol{x}}_k$ is inferred from sensor data $\boldsymbol{y}_k$ via a state estimator. This approach has been realised through Kalman filtering for linear systems (Rawlings *et al.*, 2020), Extended Kalman filtering for nonlinear systems (Lee & Ricker, 1994), and Unscented Kalman filtering (Rahideh & Shaheed, 2012). However, due to the presence of constraints or nonlinearity, the certainty equivalence (Van de Water & Willems, 1981) does not apply. It states that the optimal control policy is determined by treating the estimated state $\hat{\boldsymbol{x}}_k$ as if it were the true state, thereby separating the estimation and control problems. In constrained or nonlinear scenarios, simply applying the optimal control to the estimated state generally does not guarantee closed-loop stability. Robust output-feedback MPC formulations, as surveyed in Findeisen *et al.* (2003), are often employed when guarantees are required.

A limitation of MPC arises from its computational demand. At each control update, the cost function must be evaluated through model rollouts, and the optimisation over the prediction horizon must be repeatedly solved. This becomes particularly demanding for systems requiring long prediction horizons (e.g., systems with significant time delays) or exhibiting high-dimensional dynamics (Zafra *et al.*, 2023; Hovland *et al.*, 2006). A common strategy to mitigate this computational burden is the use of reduced-order models as extensively discussed in Section 2.4.3, which capture the essential dynamics relevant for control while significantly lowering the computational cost. Advances in hardware performance and increasingly efficient solvers for nonlinear optimisation further enhance the real-time applicability of MPC. In recent years, ROM techniques have been increasingly integrated into MPC frameworks to enable real-time control. Notable examples include formulations based on Dynamic Mode Decomposition (Lu & Zavala, 2021), Koopman operator theory (Arbabi *et al.*, 2018), Spectral Submanifolds (Alora *et al.*, 2023), and POD-based models (Hovland *et al.*, 2006; Karg & Lucia, 2021).

3.2.3. *Main advantages and limitations*

The control paradigm underlying MPC presents several characteristics that make it attractive for a wide range of applications:

- **Flexibility of model usage.** MPC can incorporate a broad spectrum of plant models, including continuous-time or discrete-time, linear or nonlinear, deterministic or stochastic representations. This modelling versatility makes it suitable for heterogeneous domain applications.
- **Natural handling of constraints.** State and input constraints can be imposed directly within the optimisation problem. This includes safety limits, actuator saturation, rate constraints, and time-delay considerations. The explicit treatment of constraints is one of the principal reasons for the rapid usage of MPC in safety-critical applications such as autonomous driving, robot motion planning, and power systems.
- **Predictive and anticipatory behaviour.** By forecasting future trajectories, MPC can anticipate upcoming events, such as setpoint changes, disturbances, or actuator interactions, and adjust its actions accordingly. This enables coordinated multi-input control strategies and improved performance over purely reactive controllers.
- **Feedback through the receding horizon mechanism.** Although the optimisation is open-loop over the prediction horizon, the repeated measurement update at every time step induces a closed-loop behaviour. This receding-horizon structure provides a degree of inherent robustness against moderate disturbances, modelling inaccuracies, and measurement noise. For small perturbations, robustness properties can even be formally established in terms of input-to-state stability.
- **Compatibility with uncertainty quantification and robust variants.** Predictive rollouts allow explicit estimation of uncertainty propagation, enabling robust MPC formulations at the cost of increased computational effort. This capability is crucial for high-stakes systems operating under uncertainty.
- **Integration with modern learning and estimation techniques.** MPC integrates naturally with online parameter estimation, adaptive modelling, observer design, and data-driven model learning.
- **Computational cost** The online implementation of MPC remains challenging due to the computational effort required to solve an optimisation problem at each control step. Higher model complexity and longer prediction horizons increase latency, while shorter horizons reduce computational cost but risk *short-sighted* or unstable control. Advances in hardware and numerical solvers are progressively enabling more efficient implementations, expanding the applicability of MPC.

These advantages underscore the considerable potential of MPC in fluid-flow control, whose main implementations are illustrated in the subsequent section.

3.3. Main implementation for fluid flows

Despite the intrinsic challenges associated with controlling fluid flows, the literature reports a number of remarkable MPC implementations across different flow scenarios.

One of the pioneering works was presented by Bewley *et al.* (2001), who applied MPC to turbulent channel flows using the DNS simulation as an exact plant model. The control objective was to reduce turbulent kinetic energy and drag through unsteady zero-net-mass-flux blowing and suction at the walls, establishing a benchmark for feedback algorithms in turbulence control. Subsequent studies evaluated MPC performance relative to alternative control strategies. The work by Fabbiane *et al.* (2014) provided a systematic assessment of MPC against both model-based and model-free approaches for delaying the laminar-turbulent transition in convectively unstable flows, highlighting the predictive advantage offered by horizon-based optimisation. In recent years, the integration of data-driven and learning-based approaches has significantly expanded applicability of MPC. Arbabi *et al.* (2018) introduced a Koopman-based MPC framework for stabilising a lid-driven cavity, leveraging linear predictors for nonlinear partial differential equations. Building on this idea, Morton *et al.* (2018) proposed DeepMPC, which combines receding-horizon optimisation with deep neural networks approximating the Koopman operator, and demonstrated its efficacy in controlling the flow past a circular cylinder. DeepMPC was further extended to more complex configurations, such as the wake of a fluidic pinball, including implementations with limited sensor measurements (Bieker *et al.*, 2020). More recently, a parsimonious, data-driven MPC framework using ROM based on Long Short-Term Memory, and a minimal set of pressure sensors, was applied to a two-dimensional truck wake (Solera-Rico *et al.*, 2025). This approach demonstrates practical, realisable MPC with limited sensing in high-dimensional flows. Another application for wake flows is the work by Williams *et al.* (2024), which employs reservoir computing as a predictive model strategy. Another line of research emphasises parsimonious, low-data models for MPC. Kaiser *et al.* (2018) demonstrated that SINDy provides compact system representations, which can improve MPC performance in low-data scenarios.

Beyond canonical flow configurations, MPC has been applied to trajectory optimisation for mobile sensors navigating unsteady flow fields (Krishna *et al.*, 2022), as well as to compressible flows where MPC was compared with neural-network controllers across multiple systems (Déda *et al.*, 2023). These studies illustrate the adaptability of MPC to both fixed and mobile actuator setups, demonstrating predictive and anticipatory capabilities even under partial observability and real-time constraints.

Overall, these examples highlight that, despite the computational and modelling challenges inherent to fluid flows, MPC has been successfully implemented across a wide spectrum of scenarios. The combination of model-based predictions, horizon optimisation, and modern data-driven methods has enabled robust and efficient control, paving the way for future applications in high-dimensional, nonlinear, and partially observed flow systems.

3.4. Towards learning in Model Predictive Control

Over the past years, the traditional paradigm of MPC, based on fixed, pre-identified models and constraints, has been extended to incorporate learning directly within the control loop. By allowing the controller to adapt its model, cost function, or other components based on operational data, this paradigm addresses limitations of static formulations, enabling significantly improved performance in complex, uncertain, and time-varying environments. This approach is philosophically aligned with the core concept that *"learning from interaction with the environment is a foundational idea underlying nearly all theories of learning and intelligence,"* as emphasised by Sutton & Barto (2018).

Building on this idea, learning-based MPC methods have been developed along several complementary directions, which can be broadly categorised into three main classes, as extensively discussed by Hewing *et al.* (2020).

The first category focuses on *learning the system dynamics*, which is the core component of the algorithm. This learning can occur during real-time operation (online learning) or across multiple deployments of the controller (adaptation). System identification may be conducted passively, by observing the natural behaviour of the system, or actively, by intentionally exciting the inputs to improve knowledge of the dynamics. This latter approach relates to the classical concept of dual control (Wittenmark, 1995), which addresses the simultaneous challenge of system identification and optimal control. Dual control was originally introduced by Feldbaum (1960), and a comprehensive survey on its application in MPC is provided by Mesbah (2018).

The second category exploits MPC primarily to provide *safety guarantees*. In this framework, MPC ensures constraint satisfaction and safe operation, while a separate learning-based controller, such as one based on RL, is responsible for performance optimisation. This decoupling allows the learning component to focus on improving control objectives without violating safety requirements (Koller *et al.*, 2018; Wabersich *et al.*, 2021).

The third category, which is the most relevant in this dissertation, addresses *learning the controller design*. Here, the focus shifts from predicting the system dynamics to optimising the parameters of the MPC formulation itself, such as the cost function weights, prediction horizon, or terminal ingredients, to enhance closed-loop performance. The motivation stems from the fact that the relationship between a specific parametrisation of MPC and the resulting closed-loop behaviour is typically complex and difficult to describe analytically (Hewing *et al.*, 2020). Many approaches have used optimisation strategies like Bayesian Optimisation (BO) to automatically select the parameters of MPC using closed-loop control realisations (Berkenkamp *et al.*, 2016; Neumann–Brosig *et al.*, 2020). In **Paper 1** of this dissertation MPC is equipped with a parametrised cost function, including weights and prediction horizon, and BO is employed to automatically tune these parameters based on closed-loop performance data. This strategy minimises manual intervention and systematically improves controller performance in complex wake flow scenarios.

3.5. The connections with Reinforcement Learning

As discussed throughout this chapter, MPC relies on a recurrent optimisation procedure in which a cost function is evaluated through repeated rollouts of a model of the plant. From a theoretical perspective, this procedure can be interpreted as a form of value-based dynamic programming: the rollouts provide estimates of the cost-to-go, which are then used to select the optimal control action over the receding horizon.

This observation highlights a natural connection between MPC and RL. Both frameworks aim to optimise future performance by estimating the expected cost or reward associated with system trajectories. More precisely, MPC performs online optimisation with a known model, effectively computing a local approximation of the value function at each timestep, while RL seeks to learn the value function or policy over repeated interactions with an unknown environment. Despite their historical development along largely independent paths and their differing practical motivations, the two approaches share a common foundation: they can both be interpreted under a unified framework rooted in the Bellman equation.

Recently, Bertsekas (2024) proposed a formal unification of MPC and RL under a common framework based on approximate Dynamic Programming. From this perspective, MPC can be viewed as performing a value-based optimisation, where the lookahead minimisation corresponds mathematically to a Newton step for solving the Bellman equation. This connection explains the strong theoretical similarity between MPC and RL, showing that both methods fundamentally operate by approximating and improving the cost-to-go function.

Continuing on this perspective, learning-based MPC can be seen as an explicit form of model-based RL algorithm. By adapting the model, cost function, or terminal components using data collected during operation, MPC effectively performs online policy improvement in a manner analogous to value-function updates in RL. Standard MPC, conversely, corresponds to the case of model-based RL where the dynamics model is assumed to be perfectly known and static, and the planning is performed entirely on this fixed knowledge (Luo *et al.*, 2024*b*).

Despite these conceptual similarities, MPC and RL differ in their formulation and historical development. MPC relies explicitly on a predictive model and receding-horizon optimisation, whereas RL often operates with unknown dynamics and incremental learning through interactions with the environment. Nevertheless, both frameworks share the same underlying structure: they seek to optimise the expected future cost by combining predictions of system evolution with a measure of performance. This commonality highlights that MPC can be understood as a principled, planning-oriented subset of RL, providing a foundation for hybrid approaches that leverage the strengths of both methodologies, particularly in complex and high-dimensional systems such as fluid flows.

Chapter 4

The test cases

This chapter presents the test cases which were considered in the studies included in the present dissertation. The dynamical systems used, together with their significance for the control and estimation procedures, are examined. The objective is to provide a clear rationale for selecting these and to show how they contribute to the overall assessment of the proposed methodologies. For clarity purposes, this chapter of the dissertation adopts a notation consistent with conventions commonly used in fluid dynamics. The symbols u, v, and w may denote velocity components or the states of partial differential equations, while x and y represent spatial coordinates. The actuation inputs are indicated by the vector $\mathbf{b}$.

4.1. The Fluidic Pinball

The first test case is the viscous and incompressible flow around a cluster of three cylinders with equal diameter $D = 2R$, commonly referred to as the fluidic pinball. In this configuration, initially proposed by Noack & Morzyński (2017), the centres of each cylinder are placed at the vertices of an equilateral triangle with side $3R$ and are symmetrically positioned with respect to the incoming flow, whose constant velocity is U_∞. The leftmost cylinder vertex is placed upstream, while the base side of the triangle is positioned orthogonal to the main direction of the flow. A Cartesian coordinate system (x, y, z) is used to describe the flow, where x is aligned with the incoming flow direction, y is perpendicular to x and points upwards, and z is orthogonal to both, such that (x, y, z) forms a right-handed coordinate system. The location in this reference is then described by $\boldsymbol{x} = (x, y, z) = x\boldsymbol{e}_x + y\boldsymbol{e}_y + z\boldsymbol{e}_z$ with $\boldsymbol{e}_y$, $\boldsymbol{e}_y$ and $\boldsymbol{e}_y$ the the unit vectors in the directions of the x, y and z axes respectively. The origin of the coordinate system is placed at the middle point between the centres of the downward top and bottom cylinders. The coordinates of the centres in the fluidic pinball configuration as proposed by Deng *et al.* (2018) are given as follows:

$$\begin{aligned}
\boldsymbol{x}_1 &= \left(-\frac{3}{2}D\cos\frac{\pi}{6},\ 0\right), \\
\boldsymbol{x}_2 &= \left(0,\ \frac{3}{4}D\right), \\
\boldsymbol{x}_3 &= \left(0,\ -\frac{3}{4}D\right),
\end{aligned} \tag{4.1}$$

with $1,\ 2,\ 3$ referring to the front, top and bottom cylinders, respectively. In addition, each cylinder can rotate independently about its axis at tangential velocities indicated by b_1, b_2 and b_3, respectively, and considered positive if counterclockwise. The rotation of the cylinders allows for active manipulation of the flow. A schematic of the fluidic

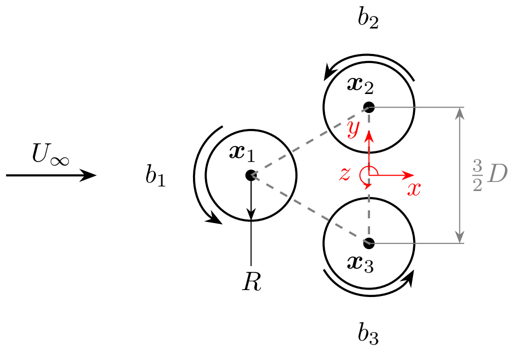

Figure 4.1: Schematic of the fluidic pinball configuration. Cylinders 1 (front), 2 (top), and 3 (bottom) are arranged at the vertices of an equilateral triangle with centres $\boldsymbol{x}_1$, $\boldsymbol{x}_2$, and $\boldsymbol{x}_3$ and radius R. The incoming flow has velocity U_∞, and each cylinder rotates independently with tangential velocities b_1, b_2, and b_3. The Cartesian reference system (axes x, y, z) is represented in red.

pinball configuration is shown in Figure 4.1. In addition, the flow field is described by the velocity vector $\boldsymbol{u} = (u, v, w)$, the pressure p, and the constant density ρ. The global aerodynamic coefficients are defined from the total forces acting on the three cylinders. The lift coefficient is defined as $C_L = \frac{2F_L}{\rho U_\infty^2 D}$, where F_L is the total lift in the y-direction. Analogously, the drag coefficient is given as $C_D = \frac{2F_D}{\rho U_\infty^2 D}$, with F_D denoting the total drag in the x-direction.

Although geometrically simple, the fluidic pinball exhibits complex wake dynamics resulting from the interaction among the rotating cylinders. This makes it a representative and versatile benchmark for wake flows behind bluff bodies, capturing behaviours that are more intricate than those observed behind a single cylinder. Depending on the Reynolds number based on the cylinder diameter, defined as $Re_D = \frac{U_\infty D}{\nu}$ with ν denoting the kinematic viscosity of the incoming flow, the wake of the fluidic pinball exhibits a wide spectrum of behaviours, from steady dynamics to transition to chaos. In addition, independent rotation of the cylinders can break the symmetry of the wake, generate lift and side forces via the Magnus effect, and produce a wide variety of dynamical patterns. The combination of multiple actuators and nonlinear flow responses makes the fluidic pinball particularly suitable for investigating Multi-Input-Multi-Output flow control strategies, where coordinated cylinder rotation can be used to actively manipulate the downstream wake and modulate the overall flow forces.

4.1.1. *Uncontrolled flow dynamics*

To characterise the unactuated flow around the fluidic pinball, the solution is analysed as the Reynolds number is progressively increased. This approach allows us to identify the successive flow regimes, the bifurcations that occur, and the associated changes in wake dynamics. The description is accompanied by a representation of the different bifurcations (Figure 4.2) that the flow undergoes as the Reynolds number increases (Deng *et al.*, 2020).

- **Steady regime**. For Reynolds number $Re_D \leq 18$ the flow exhibits a steady symmetric solution that is stable.
- **Onset of vortex shedding.** At $Re_D = 18$, the flow undergoes a Hopf bifurcation that triggers the vortex shedding phenomenon. In this regime, the lift coefficient C_L starts becoming oscillatory due to the generation of the von Kármán vortex street. Vortices are periodically and alternately shed from the top and bottom surfaces of the two rear cylinders. In addition, the gap present between the rightmost cylinders allows the creation of a base-bleeding jet whose presence is crucial in the successive bifurcations encountered at higher Reynolds numbers.
- **Symmetry breaking.** At $Re_D = 68$, a pitchfork bifurcation breaks the symmetry with respect to the axis $y = 0$. The symmetric limit cycle becomes unstable, and two mirror-conjugate asymmetric limit cycles emerge, with the gap jet deflected either upward or downward. The mean flow corresponding to each limit cycle reflects this broken symmetry. A symmetric mean flow still exists but is unstable, so the dynamics eventually settle on one of the asymmetric states as shown in Figure 4.2. This bifurcation affects both the instantaneous flow and its time-averaged structure. As a consequence of that, the mean value of the lift coefficient appears to be non-vanishing anymore due to the nonsymmetric configuration in the gap jet.
- **Quasi-periodic dynamics.** When the Reynolds number is further increased until $Re_D = 104$, the flow undergoes a Neimark-Sacker bifurcation, due to the presence of a new frequency in the power spectrum of C_L. This frequency, approximately one order of magnitude smaller than the vortex shedding frequency, modulates the amplitude of the base-bleeding jet and the main oscillator.
- **Chaotic regime.** Finally, the flow transitions to chaotic behaviour at $Re_D = 115$, where the system follows a Ruelle-Takens-Newhouse route to chaos (Deng *et al.*, 2018). Beyond this value, the lift coefficient spectrum widens significantly, and the instantaneous solutions of the flow around the fluidic pinball are characterised by a random switching between an upward or downward base-bleeding jet.

A more detailed description of the successive bifurcations of the fluidic pinball dynamics is explained by Deng *et al.* (2018) or by the diverse approach involving reduced order models with Galerkin-based (Deng *et al.*, 2020, 2021), cluster-based approaches (Deng *et al.*, 2022) or data-driven manifold learning strategies (Farzamnik *et al.*, 2023).

4.1.2. *Actuated dynamics*

The dynamics of the fluidic pinball under actuation reveal a rich set of flow responses. Depending on the rotational velocities of the three cylinders, three main control mechanisms can be identified:

- **Boat-tailing / Base-bleeding**. This mechanism is achieved by rotating the two rear cylinders in opposite directions, quantified by the parameter $p_1 = (b_3 - b_2)/2$. For $p_1 > 0$, depicted in Figure 4.3(a), the shear layers from the top and bottom cylinders are vectored towards the pinball axis, creating a boat-tailing effect. This increases the pressure behind the rear cylinders, delays separation, and energises the flow, substantially weakening or suppressing the base-bleeding jet and stabilising the wake. Conversely, for $p_1 < 0$ (Figure 4.3(b)), the gap jet is strengthened, disturbing the vortex shedding and amplifying flow oscillations. This mechanism primarily influences the drag coefficient of the fluidic pinball.

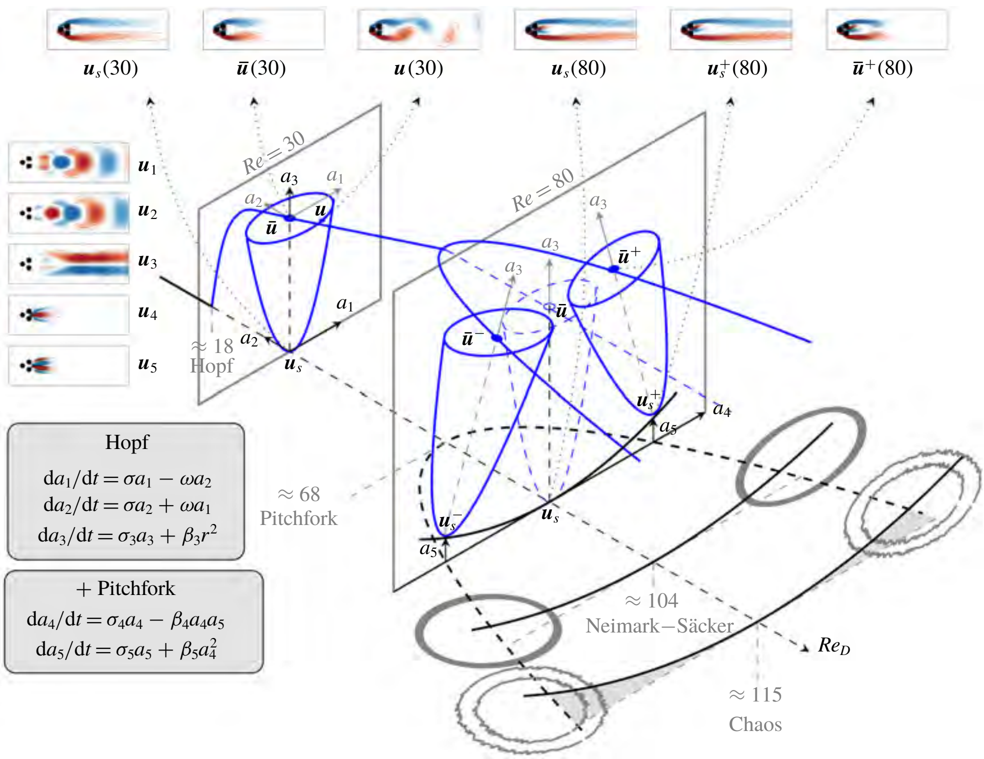

Figure 4.2: Successive bifurcations of the wake behind the fluidic pinball as the Reynolds number increases. The flow remains steady up to $Re_D = 18$, after which vortex shedding emerges. At $Re_D = 68$, a symmetry-breaking pitchfork bifurcation leads to asymmetric limit cycles. At $Re_D = 104$, the flow becomes quasi-periodic, before transitioning to chaos at $Re_D = 115$. Figure illustrating the corresponding Galerkin reduced-order model solutions in the different wake regimes. Figure reproduced with permission from Deng *et al.* (2020).

- **Magnus control**. When all three cylinders rotate synchronously (Figure 4.3(c)), the effect is quantified by $p_2 = b_1 + b_2 + b_3$. This mechanism controls the direction of the wake and regulates its symmetry, and it is mainly reflected in variations of the lift coefficient.
- **Forward stagnation point control**. Rotation of the front cylinder alone (Figure 4.3(d)) modifies the position of the stagnation point, leading to milder changes in the lift compared to Magnus control. This effect is measured by $p_3 = b_1$ and allows fine adjustment of the flow without substantially affecting the overall wake symmetry.

To better understand the dynamics of the actuated fluidic pinball, we refer to Deng *et al.* (2025); Li *et al.* (2022*a*). Numerical simulations of the flow around a fluidic pinball have been considered a benchmark case to test many control algorithms. Notable examples include optimisation strategies (Li *et al.*, 2022*b*), machine learning algorithms

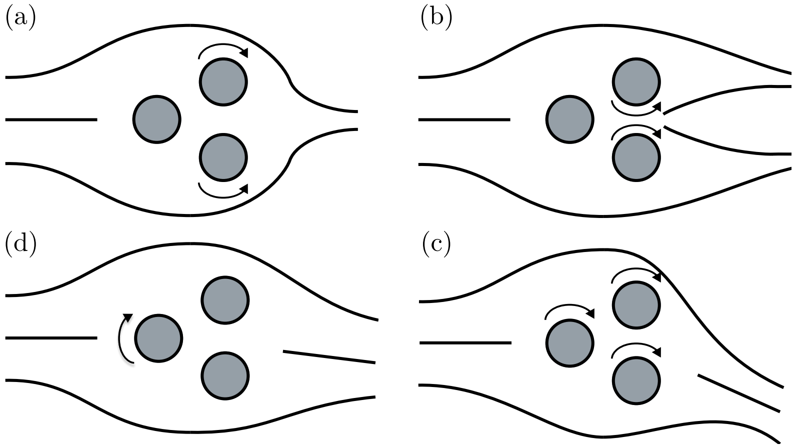

Figure 4.3: Fluidic pinball actuation mechanisms: (a) boat-tailing, (b) base-bleeding, (c) negative Magnus effect, (d) negative stagnation-point control.

(Cornejo Maceda *et al.*, 2021), deep reinforcement learning (Feng *et al.*, 2023), cluster-based approaches (Wang *et al.*, 2023) and model predictive control (Bieker *et al.*, 2020). Raibaudo *et al.* (2020) and Raibaudo & Martinuzzi (2021) also considered control in experimental scenarios.

4.1.3. *Numerical solver*

The equations that govern the flow around the fluidic pinball are the incompressible Navier-Stokes equations:

$$\begin{gathered}\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\, \mathbf{u} = -\nabla p + \frac{1}{Re_D} \nabla^2 \mathbf{u}, \\ \nabla \cdot \mathbf{u} = 0.\end{gathered} \tag{4.2}$$

where ∇ is the gradient operator, $\frac{\partial}{\partial t}$ is the partial derivative with respect to the time t and ∇^2 is the Laplace operator. The computational domain considered in this thesis is $[-6D, 20D] \times [-6D, 6D]$ while the boundary conditions comprise a far-field condition ($\boldsymbol{u} = U_\infty$) in the upper and lower edges, a stress-free one in the outflow edge and a no-slip condition on the cylinder walls, which in the absence of forcing becomes $\boldsymbol{u} = \mathbf{0}$. The DNS has been used with a discretisation of the computational domain with 8633 vertices and 4225 triangles, which proved sufficient resolution for Reynolds numbers until $Re_D = 150$ in terms of estimation errors of drag and lift coefficients (Marra *et al.*, 2024). The equations are solved by a two-dimensional DNS with the code provided by Noack & Morzyński (2017), extensively used in different works like Deng *et al.* (2020) and Farzamnik *et al.* (2023). This implements a fully implicit time integration scheme coupled with a finite-element discretisation. Time integration is third-order accurate, while the spatial discretisation employs second-order Taylor-Hood elements. Nonlinear systems are solved iteratively using a Newton-Raphson approach.

The fluidic pinball configuration has been used in the current dissertation as a test case in **Paper 1**, where it has been used as a benchmark to test the MPC architecture for drag reduction and lift control purposes. In addition, the case with $Re_D = 150$ in **Paper 3**, where it has been used for reduced order modelling applications using the ISOMAP method for $Re_D = 30$.

4.2. The Kuramoto-Sivashinsky equation

The second dynamical system considered in this dissertation is the Kuramoto-Sivashinsky equation (Kuramoto, 1978). The KS equation has been historically proposed as a model for a wide range of physical settings like chemical reaction-diffusion processes (Kuramoto & Tsuzuki, 1976), thin liquid film flows down inclined planes (Nepomnyashchii, 1974), laminar flame front instabilities (Sivashinsky, 1980; Siv Ashinsky, 1988) or the asymptotic behaviour of physical phenomena like two-phase flows in cylindrical pipes (Papageorgiou *et al.*, 1990) or plasma dynamics (Cohen *et al.*, 1976).

This dynamical system is a well-known unidimensional partial differential equation which exhibits spatio-temporally chaotic behaviour, and shares some common features with the Navier-Stokes equations (Dankowicz *et al.*, 1996), even though it does not directly model turbulence. The equation has been proposed as a canonical testbed for studying spatio-temporal chaos and control strategies in both one-dimensional and two-dimensional settings (Kalogirou *et al.*, 2015; Jiang *et al.*, 2025).

4.2.1. *The one-dimensional case*

Denoting with $u(x,t)$ the system state at time t and along the spatial coordinate x, the KS equation is given by:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + \frac{\partial^2 u}{\partial x^2} + \frac{\partial^4 u}{\partial x^4} = \phi(x,t) \tag{4.3}$$

including a nonlinear convection term, a diffusion term, a hyperdiffusion term and a forcing term in the right-hand side to enable control. The equation is integrated on a spatial domain $x \in [0, L]$ with periodic boundary conditions $u(x,t) = u(x+L,t)$. In the literature, an alternative form of the KS equation is often considered, in which the domain is fixed to $[0, 2\pi]$ and the fourth-order term is multiplied by a parameter ν, called the hyperdiffusion coefficient. Nonetheless, the two formulations are entirely equivalent under the scaling $L = 2\pi\,\nu^{-1/2}$. The choice of L (or equivalently ν) sets the characteristic spatial scale of the system, which directly affects its dynamical complexity and determines the sequence of bifurcations leading to chaos (Papageorgiou & Smyrlis, 1991). The KS system is chaotic for $L \geq 22$ (Hyman & Nicolaenko, 1986; Özalp *et al.*, 2026).

In this dissertation, a domain of length $L = 22$ is adopted, following Bucci *et al.* (2019). Under this assumption, the dynamics of the KS equation exhibit oscillations around a set of six steady state and travelling wave solutions as shown in Figure 4.4.

Regarding control, the additional forcing term in the right-hand side of the equation is modelled as a weighted sum of Gaussian actuators as

$$\phi(x,t) = \sum_{i=1}^{4} b^{(i)}(t)\,\frac{1}{\sqrt{2\pi\sigma_a^2}}\exp\left(-\frac{(x - x_a^{(i)})^2}{2\sigma_a^2}\right), \tag{4.4}$$

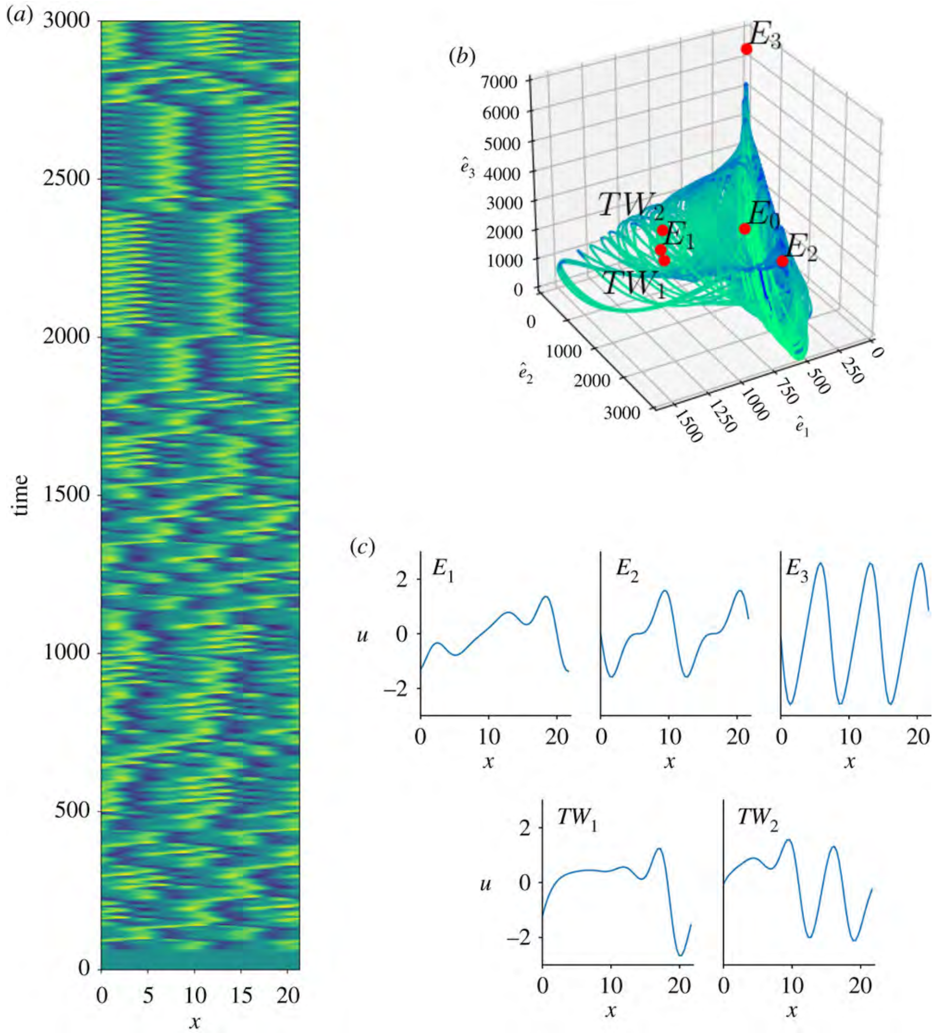

Figure 4.4: Dynamics of the Kuramoto-Sivashinsky equation over a time interval of 3000 units. (a) Space-time contour of a trajectory starting from a null perturbed state (E_0). (b) The same trajectory is projected onto the phase space spanned by the dominant Fourier coefficients ($\hat{e}_1, \hat{e}_2, \hat{e}_3$). Red markers indicate the trivial solution E_0, the three non-trivial equilibria E_i ($i = 1, 2, 3$), and the travelling waves TW_1 and TW_2. (c) Spatial profiles of the indicated solutions. Figure reproduced with permission from Bucci *et al.* (2019)

where $x_a^{(i)} = \frac{iL}{4}$ denote the actuators locations, σ_a defines the standard deviation of the Gaussian support, and $b^{(i)}(t)$ is the time-dependent amplitude of the i-th actuator. For control purposes, the sensors are considered to be uniformly distributed along the spatial domain, providing evenly spaced measurements of the system state.

The equation is integrated in the Fourier domain using a pseudo-spectral method. Specifically, the system state $v(x, t)$ is expanded in terms of Fourier modes as

$$u(x,t) = \sum_{k=-N/2}^{N/2-1} \hat{u}_k(t)\, e^{ik\frac{2\pi}{L}x},$$

where $\hat{v}_k(t)$ are the complex Fourier coefficients and N is the number of spatial discretisation points. The spatial derivatives are computed exactly in Fourier space, while the nonlinear convection term is computed in physical space and then transformed back to Fourier space (pseudo-spectral method). Time marching is carried out using a third-order semi-implicit Runge-Kutta scheme: the linear terms are treated implicitly, while the nonlinear and forcing terms are treated explicitly. For all simulations presented in this work, 64 discretisation points and a time step of 0.05 were adopted and proved sufficient resolution for integration as demonstrated by (Bucci *et al.*, 2019).

4.2.2. *The two-dimensional case*

The KS equation, as previously presented in the one-dimensional version, naturally extends to the two-dimensional case as:

$$\frac{\partial u}{\partial t} + \frac{1}{2}\left|\nabla u\right|^2 + \nabla^2 u + \nabla^2(\nabla^2 u) = 0, \tag{4.5}$$

where $u(x, y, t)$ is the system state along the two spatial coordinates x and y at time t and $\nabla^2(\nabla^2 u)$ is the bi-Laplacian term. The equation is integrated in a rectangular domain of length L_x and L_y in the two spatial coordinates, respectively. The system is subject to periodic boundary conditions:

$$u(x, y, t) = u(x + 2L_x, y, t) = u(x, y + 2L_y, t), \tag{4.6}$$

Similarly to the one-dimensional case, the dynamics is governed by nonlinear advection (gradient term), linear diffusion (Laplacian term) and hyper-diffusion (bi-Laplacian) terms. The dynamics of the equation are generally determined as the one-dimensional case by L_x and L_y. Nevertheless, the equation can be defined introducing the following coordinate transformation as adopted in Kalogirou *et al.* (2015)

$$x \to \frac{L_x}{\pi}x, \quad y \to \frac{L_y}{\pi}y, \quad t \to \frac{L_x\, L_y}{\pi^2}t, \tag{4.7}$$

which is useful to normalise the domain $[0, 2\pi] \times [0, 2\pi]$. Under this rescaling, the equation becomes:

$$\frac{\partial u}{\partial t} + \frac{1}{2}\left|\nabla_\nu u\right|^2 + \nabla_\nu^2 u + \nu_1 \nabla_\nu^2(\nabla_\nu^2 u) = 0, \tag{4.8}$$

with

$$\nabla_\nu = \left(\frac{\partial}{\partial x}, \frac{\nu_2}{\nu_1}\frac{\partial}{\partial y}\right), \quad \nabla_\nu^2 = \frac{\partial^2}{\partial x^2} + \frac{\nu_2}{\nu_1}\frac{\partial^2}{\partial y^2}, \tag{4.9}$$

and

$$\nu_1 = \left(\frac{\pi}{L_x}\right)^2, \quad \nu_2 = \left(\frac{\pi}{L_y}\right)^2. \tag{4.10}$$

In addition, to enable control, a forcing term is added in the right-hand side of Equation (4.8). It is defined as:

$$\Gamma(x, y, t) = \frac{1}{2\pi\sigma_a^2} \sum_{j=1}^{n_u} b^{(j)}(t) \exp\left(-\frac{(x - x_a^{(j)})^2 + (y - y_a^{(j)})^2}{2\sigma_a^2}\right), \tag{4.11}$$

where $b^{(j)}(t)$ is the time-dependent amplitude of the jth Gaussian actuator located at $(x_a^{(j)}, y_a^{(j)})$ and σ_a denotes the standard deviation of the Gaussian support.

The selection of the scaling parameters ν_1 and ν_2 determines the behaviour of the dynamical system as shown in Figure 4.5, from steady states to chaotic solutions. The two-dimensional KS equation is integrated in the Fourier domain using a spectral Galerkin method. The solution $u(x, y, t)$ is expanded in a tensor-product Fourier basis, which allows exact evaluation of all spatial derivatives, including the gradient, Laplacian, and bi-Laplacian. The nonlinear term $\frac{1}{2}|\nabla_\nu u|^2$ is treated pseudo-spectrally: the field is transformed to physical space, the nonlinear products are evaluated pointwise, and then transformed back to Fourier space. Time integration is carried out using a fourth-order exponential Runge-Kutta scheme, which efficiently handles the stiffness introduced by the fourth-order spatial derivatives while maintaining stability.

In this dissertation, the equation with $\nu_1 = 0.3$ and $\nu_2 = 0.16$ is presented. In this setting, the system exhibits a modulated periodic regime, alternating between stable travelling waves and oscillatory phases with repeated, identical amplitude patterns. The equation is integrated on a computational grid of 32×32 points with a time step of $\Delta t = 0.0025$. This spatial and temporal resolution was verified to be sufficient through comparison with two reference simulations: one using a 64×64 grid with the same time step, which showed differences of order $O(10^{-5})$, and another using the same 32×32 grid with a smaller time step $\Delta t = 0.001$, which exhibited errors of order $O(10^{-3})$.

The only studies applying the two-dimensional KS equation for control-related purposes are Kang & Fridman (2021), where sampled-data control is used to stabilise the null solution, and Jiang *et al.* (2025), where deep RL is used to improve convergence of a numerical solver and to explore trajectory optimisation between fixed points of the equation. This test case is used as a benchmark to test the latent-space MPC on spatially extended systems with nonlinear dynamics in **Paper 2**.

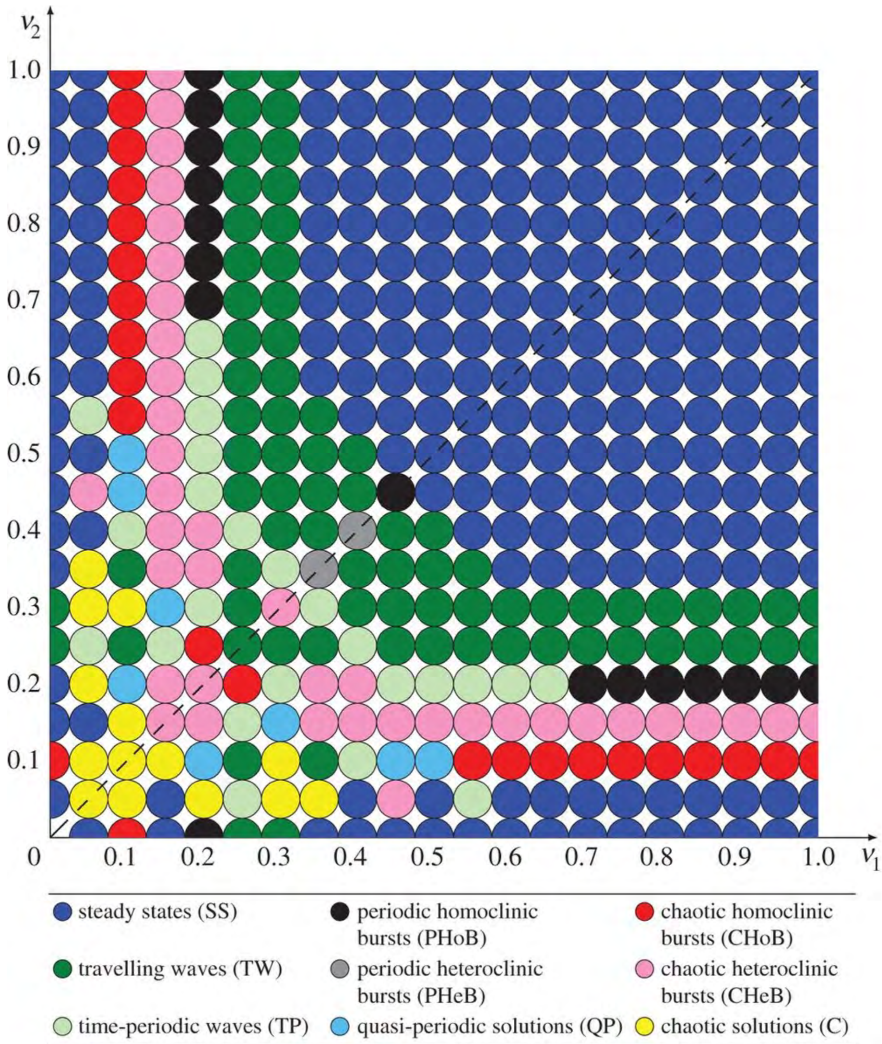

Figure 4.5: Dynamical regimes of the two-dimensional Kuramoto-Sivashinsky equation for different combinations of the parameters ν_1 and ν_2. Starting from the reference case $\nu_1 = \nu_2 = 1$, varying these parameters produces a wide range of behaviours, from steady states to fully chaotic spatiotemporal dynamics. Figure reproduced with permission from Kalogirou *et al.* (2015).

Chapter 5

Conclusions and future research

This chapter presents the main contribution of the dissertation, summarises the key highlights of the included articles, and discusses potential future developments arising from this work.

5.1. Main contributions

The core motivation of this dissertation stems from the challenge of applying model predictive control to fluid flows. While it offers a powerful framework for optimisation under constraints and inherently robust control, its implementation is hindered by the high dimensionality, strong nonlinearity, and partial observability characteristic of many problems arising in fluid dynamics. This motivates the development of strategies for efficient system identification, particularly through reduced-order modelling and robust state estimation from sparse, noisy measurements. The main contribution of this work lies in advancing these methods to bring model predictive control closer to real-time, practical application in complex fluid systems.

The first contribution (**Paper 1**) addresses the challenges of model identification and practical tuning in the context of wake flows, which are critical in aerodynamics. The work proposes effective control by directly modelling the aerodynamic forces on the body, thereby bypassing the need for computationally expensive, full fluid state estimation. This approach effectively exploits the intrinsic sparse nature that dynamic systems often exhibit in feature spaces. Furthermore, this contribution addresses the complex problem of performing controller tuning, especially in the presence of multi-objective optimisation inherent in advanced control schemes. It introduces a learning-based model predictive control strategy for automatic parameter selection, significantly reducing the need for intensive user intervention while optimising closed-loop performance. This paper also tackles the mitigation of measurement noise effects by employing statistical techniques for accurate force estimation, ensuring the robustness of the predictive model.

The second contribution (**Paper 2**) focuses on mitigating the computational burden of the control strategy, particularly for spatially extended systems. This research provides a comprehensive framework for control directly defined within a latent representation of the physical space, identified using modal decomposition techniques. The predictive model in this framework provides predictions on the future state evolution of the flow directly in a reduced-order representation of the plant dynamics. This allows the identification of control strategies entirely within a low-dimensional space, significantly increasing the applicability of MPC when the evaluation of the plant trajectories is computationally intensive. Crucially, this work also addresses the fundamental problem of partial observability: when sparse, incomplete, and noisy feedback is provided to the controller, accurate state reconstruction is necessary. The framework explores the usage of an estimation layer based on a nonlinear Kalman filter for accurate state reconstruction for control. The resulting methodology proves successful on surrogate test cases exhibiting characteristics such as spatial extension and nonlinearities, similar to those found in fluid flows.

The final contribution (**Paper 3**) addresses the fundamental challenge of model structure identification for controlled fluid flows. This work develops a methodology to identify coordinates that capture the dominant, controllable dynamics of a flow in the presence of steady actuation. Using post-transient snapshot data, the approach constructs a low-dimensional subspace, termed the actuation manifold, which encodes the directions in state space most affected by control inputs. By projecting the system dynamics onto this manifold, it becomes possible to design efficient, reduced-order control strategies without the need for full high-dimensional simulations. The identified coordinates are shown to align closely with the physical structures governing the flow response, providing a physically interpretable and computationally efficient basis for future integration into model predictive control frameworks.

5.2. Paper highlights

Paper 1: Marra, L., Meilán-Vila, A., & Discetti, S. 2024. Self-tuning model predictive control for wake flows. *J. Fluid Mech.*, 983, A26. DOI: 10.1017/jfm.2024.47.

- A reduced-order model is obtained via sparse identification of nonlinear dynamics directly from aerodynamic force coefficients.
- A model predictive control scheme is implemented based on the identified data-driven force model.
- Forces are directly measured with noise contamination, and estimation is performed using local polynomial regression.
- Control hyperparameters are optimised using Bayesian Optimisation.
- The optimised model predictive control algorithm significantly outperforms manually tuned counterparts and reduces the need for user intervention, enabling autonomous and efficient control.
- The approach is exemplified on the two-dimensional laminar chaotic wake past the fluidic pinball at Reynolds number $Re = 150$.

Paper 2: Marra, L., Semeraro, O., Mathelin, L., Meilán-Vila, A., & Discetti, S. 2025. Latent-Space Non-Linear Model Predictive Control for Partially Observable Systems. *arXiv preprint.* DOI: arXiv.2511.19056 (Under review)

- A latent space is identified from Proper Orthogonal Decomposition modes extracted from data.
- A predictive model is obtained in this latent space using Operator Inference.
- A model-based control framework is defined in the latent space to reduce computational cost.
- State estimation in the latent space is performed using an Unscented Kalman Filter from sparse and noisy measurements.
- The approach is exemplified on the one- and two-dimensional Kuramoto-Sivashinsky equations, which serve as benchmarks for nonlinear, chaotic, and spatially extended systems.

Paper 3.: Marra, L., Cornejo Maceda, G. Y., Meilán-Vila, A., Guerrero, V., Rashwan, S., Noack, B. R., Discetti, S. & Ianiro, A. 2024. Actuation manifold from snapshot data. *J. Fluid Mech.*, 996, A26. DOI: 10.1017/jfm.2024.593.

- An encoding strategy based on isometric mapping is employed to identify a data-driven actuation manifold representing the post-transient response of wake flows.
- The discovered coordinates are low-dimensional and strongly aligned with physical parameters, such as actuation and wake response.
- A neural-network estimator is proposed to obtain the low-dimensional representation using only sparse measurements and actuation information.
- A decoding strategy reconstructs the full flow snapshots from the latent representation using k-Nearest-Neighbour interpolation.
- The approach is exemplified on the two-dimensional periodic wake past the fluidic pinball at Reynolds number $Re = 30$.

5.3. Perspective

The discussed topics and proposed methodologies pave the way for several natural extensions of the present work, as proposed below.

- **Experimental validation of Paper 1**: The developments in Model Predictive Control for wake flows naturally suggest a transition from numerical to experimental validation. The candidate has actively contributed to the design of a dedicated experimental facility, enabling assessment of the self-tuning controller under real fluid dynamics conditions. This extension will verify the robustness of the data-driven force model and the automated tuning strategy when confronted with measurement noise and unmodelled dynamics in real experimental scenarios. A challenge in applying this method concerns the modelling strategy. A benchmark between the SINDy-based approach and alternative architectures would be desirable. Methods to be evaluated include input-output mapping via Recurrent Neural Networks. Different state embeddings could also be assessed, as instantaneous observables might be insufficient when the system dimensionality increases. To extend the method to turbulence control, the state definition may need to be enriched using delay-coordinate embeddings, among others.
- **Nonlinear latent-space methods for Paper 2**: The latent-space predictive control framework can be extended using nonlinear dimensionality reduction techniques, such as autoencoders (see Section 2.4.2), to achieve more compact latent representations than the linear methods used in this dissertation. This compression can capture additional nonlinear correlations in the system, allowing for more efficient and expressive reduced-order models. Such enhanced representations naturally enable the extension of model predictive control to more complex, partially observable flow scenarios, where linear latent spaces may be insufficient to encode the dominant dynamics for accurate prediction and control. These nonlinear compression methods allow the extraction of low-dimensional latent dynamics, improving predictive efficiency and identifying effective control coordinates for implementing MPC. Extending the framework to nonlinear autoencoders provides a practical pathway toward predictive control of turbulent flows.

- **Robust Model Predictive Control:** Both **Paper 1** and **Paper 2** provide a foundation for incorporating explicit robustness against external disturbances and measurement noise that might arise in experimental applications, for instance. Extending the frameworks in this direction will improve the reliability and safety of control strategies in complex fluid systems, where uncertainties are unavoidable.
- **Hybrid Model Predictive Control and Reinforcement Learning:** The limitations of model-based predictive control in highly complex or partially unknown environments motivate a hybrid approach with Reinforcement Learning. Model Predictive Control can govern large-scale system behaviour, ensuring safety and compliance with constraints, while Reinforcement Learning targets small-scale, fine-grained dynamics that are difficult to model. By leveraging the predictive knowledge embedded in Model Predictive Control, Reinforcement Learning can learn more efficiently, reducing sample complexity. This extension should be viewed as a strategy for tackling turbulence by combining a model-based approach with model-free Reinforcement Learning, where policies are directly parameterised, potentially via neural networks, allowing for fast, expressive, and flexible deployment.
- **Time-varying actuation for Paper 3**: The actuation manifold methodology suggests natural extensions to flows subject to time-dependent forcing. By learning the dynamic response in the Isometric Mapping (or related manifold learning) space under varying actuation, it becomes possible to construct predictive models whose state evolution is expressed entirely in the learned manifold coordinates. The resulting reduced-order dynamics enable the optimal control problem to be posed and solved directly in the manifold space. Furthermore, because the Isometric Mapping identifies a physically interpretable latent space, Model Predictive Control can be formulated directly in terms of these meaningful coordinates, enabling control laws that are both computationally efficient and physically grounded.

Bibliography

Abdufattokhov, S., Zanon, M. & Bemporad, A. 2021 Learning Convex Terminal Costs for Complexity Reduction in MPC. In *2021 60th IEEE Conf. Decis. Control*, pp. 2163–2168.

Air Transport Action Group (ATAG) 2025 Climate action. Accessed: 12-Dec-2025.

Alora, J. I., Pabon, L. A., Köhler, J., Cenedese, M., Schmerling, E., Zeilinger, M. N., Haller, G. & Pavone, M. 2023 Robust Nonlinear Reduced-Order Model Predictive Control. In *2023 62nd IEEE Conf. Decis. Control*, pp. 4798–4805.

Amendola, M., Arcucci, R., Mottet, L., Casas, C. Q., Fan, S., Pain, C., Linden, P. & Guo, Y.-K. 2021 Data assimilation in the latent space of a convolutional autoencoder. In *ICCS*, pp. 373–386. Springer.

Amico, E., Cafiero, G. & Iuso, G. 2022 Deep reinforcement learning for active control of a three-dimensional bluff body wake. *Phys. Fluids* **34** (10), 105126.

Amico, E., Montagner, S., Serpieri, J. & Cafiero, G. 2025 An a posteriori data-driven method for phase-averaging optical measurements. *Meas. Sci. Technol.* **36**, 075301.

Anderson, B. D. O. & Moore, J. B. 1990 Optimal control: linear quadratic methods. USA: Prentice-Hall, Inc.

Andrei, N. 2006 Modern control theory-a historical perspective. *Stud. Inform. Control* **10** (1), 51–62.

Arbabi, H., Korda, M. & Mezić, I. 2018 A data-driven koopman model predictive control framework for nonlinear partial differential equations. *IEEE Conf. Decis. Control* pp. 6409–6414.

Arias, P., Randall, G. & Sapiro, G. 2007 Connecting the out-of-sample and pre-image problems in kernel methods. In *CVPR*, pp. 1–8.

Arif, O., Vela, P. & Daley, W. 2010 Pre-image problem in manifold learning and dimensional reduction methods. In *ICMLA*, pp. 921–924. IEEE.

Asch, M., Bocquet, M. & Nodet, M. 2016 Data Assimilation, 1st edn. Philadelphia, PA: Society for Industrial and Applied Mathematics.

Bagheri, S., Brandt, L. & Henningson, D. S. 2009*a* Input–output analysis, model reduction and control of the flat-plate boundary layer. *J. Fluid Mech.* **620**, 263–298.

Bagheri, S., Henningson, D. S., Hœpffner, J. & Schmid, P. J. 2009*b* Input-output analysis and control design applied to a linear model of spatially developing flows. *Appl. Mech. Rev.* **62** (2), 020803.

Bakarji, J., Champion, K., Kutz, N. J. & Brunton, S. L. 2023 Discovering governing equations from partial measurements with deep delay autoencoders. *Proc. R. Soc. Math. Phys. Eng. Sci.* **479** (2276), 20230422.

Bakir, G. H., Weston, J. & Schölkopf, B. 2004 Learning to find pre-images. *Adv. Neural Inf. Process. Syst.* **16** (449-456), 1.

Bannister, R. N. 2017 A review of operational methods of variational and ensemble-variational data assimilation. *Q. J. R. Meteorol. Soc.* **143** (703), 607–633.

Bartenhagen, C., Klein, H.-U., Ruckert, C., Jiang, X. & Dugas, M. 2010 Comparative study of unsupervised dimension reduction techniques for the visualization of microarray gene expression data. *BMC Bioinform.* **11** (1), 567.

Beiki, A. & Kamali, R. 2025 Dynamic feature capturing in a fluid flow reduced-order model using attention-augmented autoencoders. *Eng. Appl. Artif. Intell.* **149**, 110463.

Belkin, M. & Niyogi, P. 2001 Laplacian eigenmaps and spectral techniques for embedding and clustering. *Adv. Neural. Inf. Process. Syst.* **14**.

Bellman, R. E. 1954*a* Dynamic programming and a new formalism in the calculus of variations. *PNAS* **40** (4), 231–235.

Bellman, R. E. 1954*b* The theory of dynamic programming. *Bull. Amer. Math. Soc.* **60** (6), 503–515.

Bellman, R. E. 1957 Dynamic Programming. Princeton, NJ, USA: Princeton University Press.

Bemporad, A. & Morari, M. 2007 Robust model predictive control: A survey. In *Robustness in identification and control*, pp. 207–226. Springer.

Benner, P., Goyal, P., Heiland, J. & Duff, I. P. 2020*a* Operator inference and physics-informed learning of low-dimensional models for incompressible flows. *arXiv preprint* .

Benner, P., Goyal, P., Kramer, B., Peherstorfer, B. & Willcox, K. 2020*b* Operator inference for non-intrusive model reduction of systems with non-polynomial nonlinear terms. *Comput. Methods Appl. Mech. Eng.* **372**, 113433.

Berkenkamp, F., Schoellig, A. P. & Krause, A. 2016 Safe controller optimization for quadrotors with Gaussian processes. In *2016 IEEE Int. Conf. Robot. Autom.*, pp. 491–496.

Berkooz, G., Holmes, P. & Lumley, J. L. 1993 The proper orthogonal decomposition in the analysis of turbulent flows. *Annu. Rev. Fluid Mech.* **25**, 539–575.

Berkovitz, L. D. 2013 Optimal control theory, *Applied Mathematical Sciences*, vol. 12. Springer Science & Business Media.

Bertsekas, D. P. 2005 Dynamic programming and suboptimal control: A survey from adp to mpc. *Eur. J. Control* **11** (4), 310–334.

Bertsekas, D. P. 2012 Dynamic Programming and Optimal Control: Volume I. *Athena Scientific Optimization and Computation Series* 3. Athena Scientific Belmont.

Bertsekas, D. P. 2024 Model predictive control and reinforcement learning: A unified framework based on dynamic programming. *IFAC-PapersOnLine* **58** (18), 363–383.

Bewley, T. R. & Liu, S. 1998 Optimal and robust control and estimation of linear paths to transition. *J. Fluid Mech.* **365**, 305–349.

Bewley, T. R., Moin, P. & Temam, R. 2001 DNS-based predictive control of turbulence: an optimal benchmark for feedback algorithms. *J. Fluid Mech.* **447**, 179–225.

Bieker, K., Peitz, S., Brunton, S. L., Kutz, J. N. & Dellnitz, M. 2020 Deep model predictive flow control with limited sensor data and online learning. *Theor. Comput. Fluid Dyn.* **34** (4), 577–591.

Billings, S. A. 2013 Nonlinear system identification: NARMAX methods in the time, frequency, and spatio-temporal domains. John Wiley & Sons.

Box, G., Jenkins, G. M., Reinsel, G. C. & Ljung, G. M. 2015 Time series analysis: forecasting and control. John Wiley & Sons.

Box, G. E. P. 1979 Robustness in the strategy of scientific model building. In *Robustness in statistics*, pp. 201–236. Elsevier.

Boyd, S., Chua, L. O. & Desoer, C. A. 1984 Analytical foundations of volterra series. *IMA J. Math. Control Inf.* **1** (3), 243–282.

Brindise, M. C. & Vlachos, P. P. 2017 Proper orthogonal decomposition truncation method for data denoising and order reduction. *Exp. Fluids* **58** (4), 28.

BRUNTON, S. L. & NOACK, B. R. 2015 Closed-loop turbulence control: Progress and challenges. *Appl. Mech. Rev.* **67** (5), 050801.

BRUNTON, S. L., PROCTOR, J. L. & KUTZ, J. N. 2016*a* Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proc. Natl. Acad. Sci.* **113** (15), 3932–3937.

BRUNTON, S. L., PROCTOR, J. L. & KUTZ, J. N. 2016*b* Sparse identification of nonlinear dynamics with control (SINDYc). *IFAC-PapersOnLine* **49** (18), 710–715, NOLCOS.

BUCCI, M. A., SEMERARO, O., ALLAUZEN, A., WISNIEWSKI, G., CORDIER, L. & MATHELIN, L. 2019 Control of chaotic systems by deep reinforcement learning. *Proc. R. Soc. A* **475** (2231), 20190351.

BUDNINSKIY, M., YIN, G., FENG, L., TONG, Y. & DESBRUN, M. 2019 Parallel transport unfolding: A connection-based manifold learning approach. *SIAM J. Appl. Alg. Geom.* **3** (2), 266–291.

BUIZZA, C., QUILODRÁN CASAS, C., NADLER, P., MACK, J., MARRONE, S., TITUS, Z., LE CORNEC, C., HEYLEN, E., DUR, T., BACA RUIZ, L., HEANEY, C., DÍAZ LOPEZ, J. A., KUMAR, K. & ARCUCCI, R. 2022 Data learning: Integrating data assimilation and machine learning. *J. Comput. Sci.* **58**, 101525.

CAMACHO, E. F. & BORDONS, C. 2007 Model Predictive Control. *Advanced Textbooks in Control and Signal Processing* 2. Springer London.

CAMPO, P. J. & MORARI, M. 1987 Robust model predictive control. In *1987 American control conference*, pp. 1021–1026. IEEE.

CASTELLANOS, R., CORNEJO MACEDA, G. Y., DE LA FUENTE, I., NOACK, B. R., IANIRO, A. & DISCETTI, S. 2022 Machine-learning flow control with few sensor feedback and measurement noise. *Phys. Fluids* **34** (4), 047118.

CHARTRAND, R. 2011 Numerical differentiation of noisy, nonsmooth data. *ISRN* **2011** (1), 164564.

CHATZIMANOLAKIS, M., WEBER, P. & KOUMOUTSAKOS, P. 2024 Learning in two dimensions and controlling in three: Generalizable drag reduction strategies for flows past circular cylinders through deep reinforcement learning. *Phys. Rev. Fluids* **9** (4), 043902.

CHENG, S., QUILODRÁN-CASAS, C., OUALA, S., FARCHI, A., LIU, C., TANDEO, P., FABLET, R., LUCOR, D., IOOSS, B., BRAJARD, J., XIAO, D., JANJIC, T., DING, W., GUO, Y., CARRASSI, A., BOCQUET, M. & ARCUCCI, R. 2023 Machine learning with data assimilation and uncertainty quantification for dynamical systems: A review. *IEEE/CAA J. Autom. Sin.* **10** (6), 1361–1387.

CHOI, H. & CHOI, S. 2007 Robust kernel isomap. *Pattern Recognition* **40** (3), 853–862.

COHEN, B. I., KROMMES, J., TANG, W. & ROSENBLUTH, M. 1976 Non-linear saturation of the dissipative trapped-ion mode by mode coupling. *Nucl. Fusion* **16** (6), 971.

CORNEJO MACEDA, G. Y., LI, Y., LUSSEYRAN, F., MORZYŃSKI, M. & NOACK, B. R. 2021 Stabilization of the fluidic pinball with gradient-enriched machine learning control. *J. Fluid Mech.* **917**, A42.

CUTLER, C. R. & RAMAKER, B. L. 1980 Dynamic matrix control– A computer control algorithm. In *JACC*, p. 72.

DANKOWICZ, H., HOLMES, P., BERKOOZ, G. & ELEZGARAY, J. 1996 Local models of spatio-temporally complex fields. *Physica D* **90** (4), 387–407.

DÉDA, T., WOLF, W. R. & DAWSON, S. T. 2023 Backpropagation of neural network dynamical models applied to flow control. *Theor. Comput. Fluid Dyn* **37** (1), 35–59.

DEE, D. P. 2005 Bias and data assimilation. *Q. J. Roy. Meteor. Soc.* **131** (613), 3323–3343.

DENG, N., NOACK, B. R., CORNEJO MACEDA, G. Y. & PASTUR, L. R. 2025 Nonlinear dynamics of the actuated fluidic pinball — steady, periodic, and chaotic regimes. *Chaos Soliton. Fract.* **193**, 116075.

Deng, N., Noack, B. R., Morzyński, M. & Pastur, L. R. 2020 Low-order model for successive bifurcations of the fluidic pinball. *J. Fluid Mech.* **884**, A37.

Deng, N., Noack, B. R., Morzyński, M. & Pastur, L. R. 2021 Galerkin force model for transient and post-transient dynamics of the fluidic pinball. *J. Fluid Mech.* **918**, A4.

Deng, N., Noack, B. R., Morzyński, M. & Pastur, L. R. 2022 Cluster-based hierarchical network model of the fluidic pinball – cartographing transient and post-transient, multi-frequency, multi-attractor behaviour. *J. Fluid Mech.* **934**, A24.

Deng, N., Pastur, L. R., Morzyński, M. & Noack, B. R. 2018 Route to chaos in the fluidic pinball **1**, V001T01A005.

Ding, Y., Wang, L., Li, Y. & Li, D. 2018 Model predictive control and its application in agriculture: A review. *Comput. Electron. Agric.* **151**, 104–117.

Dong, X., Hong, H., Deng, X., Zhong, W. & Hu, G. 2023 Surrogate model-based deep reinforcement learning for experimental study of active flow control of circular cylinder. *Phys. Fluids* **35** (10), 105147.

Dorato, P. 2003 A historical review of robust control. *IEEE Control Syst. Mag.* **7** (2), 44–47.

Doyle, J., Glover, K., Khargonekar, P. & Francis, B. 1988 State-space solutions to standard H_2 and H_∞ control problems. In *1988 American Control Conference*, pp. 1691–1696.

Duriez, T., Brunton, S. L. & Noack, B. R. 2017 Machine Learning Control – Taming Nonlinear Dynamics and Turbulence, *Fluid Mechanics and Its Applications*, vol. 116. Springer Cham.

Fabbiane, N., Semeraro, O., Bagheri, S. & Henningson, D. S. 2014 Adaptive and model-based control theory applied to convectively unstable flows. *Appl. Mech. Rev.* **66** (6), 060801.

Fan, D., Yang, L., Wang, Z., Triantafyllou, M. S. & Karniadakis, G. E. 2020 Reinforcement learning for bluff body active flow control in experiments and simulations. *PNAS* **117** (42), 26091–26098.

Fan, J. & Gijbels, I. 1996 Local polynomial modelling and its applications. *Monographs on statistics and applied probability series* 66. London: Chapman & Hall.

Farzamnik, E., Ianiro, A., Discetti, S., Deng, N., Oberleithner, K., Noack, B. R. & Guerrero, V. 2023 From snapshots to manifolds–a tale of shear flows. *J. Fluid Mech.* **955**, A34.

Fasel, U., Kutz, J. N., Brunton, B. W. & Brunton, S. L. 2022 Ensemble-sindy: Robust sparse model discovery in the low-data, high-noise limit, with active learning and control. *Proc. R. Soc. A* **478** (2260), 20210904.

Feldbaum, A. A. 1960 Dual control theory. i. *Avtomatika i Telemekhanika* **21** (9), 1240–1249.

Feng, H., Wang, Y., Xiang, H., Jin, Z. & Fan, D. 2023 How to control hydrodynamic force on fluidic pinball via deep reinforcement learning. *Phys. Fluids* **35** (4), 045137.

Findeisen, R., Imsland, L., Allgower, F. & Foss, B. A. 2003 State and output feedback nonlinear model predictive control: An overview. *Eur. J. Control.* **9** (2-3), 190–206.

Fix, E. & Hodges, J. L. 1989 Discriminatory analysis. Nonparametric Discrimination: Consistency properties. *Int. Stat. Rev.* **57** (3), 238–247.

Fletcher, C. A. J. 1984 Computational Galerkin Methods, pp. 72–85. Berlin, Heidelberg: Springer Berlin Heidelberg.

Floyd, R. W. 1962 Algorithm 97: Shortest path. *Commun. ACM* **5** (6), 345.

Fukami, K., Murata, T., Zhang, K. & Fukagata, K. 2021 Sparse identification of nonlinear dynamics with low-dimensionalized flow representations. *J. Fluid Mech.* **926**, A10.

Fukami, K., Nakamura, T. & Fukagata, K. 2020 Convolutional neural network based hierarchical autoencoder for nonlinear mode decomposition of fluid field data. *Phys. Fluids* **32** (9), 095110.

Fukami, K. & Taira, K. 2023 Grasping extreme aerodynamics on a low-dimensional manifold. *Nat. Commun.* **14** (1), 6480.

Gajamannage, K., Paffenroth, R. & Bollt, E. M. 2019 A nonlinear dimensionality reduction framework using smooth geodesics. *Pattern Recognition* **87**, 226–236.

Gao, L. M. & Kutz, J. N. 2024 Bayesian autoencoders for data-driven discovery of coordinates, governing equations and fundamental constants. *Proc. R. Soc. Math. Phys. Eng. Sci.* **480** (2286), 20230506.

Garcia, C. E. & Morshedi, A. 1986 Quadratic programming solution of dynamic matrix control (QDMC). *Chem. Eng. Commun.* **46** (1-3), 73–87.

Garcia, C. E., Prett, D. M. & Morari, M. 1989 Model predictive control: Theory and practice—a survey. *Automatica* **25** (3), 335–348.

Garcia, X., Miró, A., Suárez, P., Alcántara-Ávila, F., Rabault, J., Font, B., Lehmkuhl, O. & Vinuesa, R. 2025 Deep-reinforcement-learning-based separation control in a two-dimensional airfoil. *Int. J. Heat Fluid Flow* **116**, 109913.

Guo, X., Li, W. & Iorio, F. 2016 Convolutional Neural Networks for Steady Flow Approximation. In *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, *KDD '16*, p. 481–490. New York, NY, USA: Association for Computing Machinery.

Gad-el Hak, M. 1996 Modern developments in flow control. *Appl. Mech. Rev.* **49** (7), 365–379.

Gad-el Hak, M. & Pollard, A. ed. 1998 Flow Control: Fundamentals and Practices. *Lecture Notes in Physics Monographs* 1. Berlin, Heidelberg: Springer.

Hasegawa, K., Fukami, K., Murata, T. & Fukagata, K. 2020 CNN-LSTM based reduced order modeling of two-dimensional unsteady flows around a circular cylinder at different Reynolds numbers. *Fluid Dyn. Res.* **52** (6), 065501.

Hausknecht, M. J. & Stone, P. 2015 Deep Recurrent Q-Learning for Partially Observable MDPs. AAAI **45**, 141.

Heaney, C. E., Tang, J., Yan, J., Guo, D., Ipock, J., Kaluvakollu, S., Lin, Y., Shao, D., Chen, B., Mottet, L., Kumar, P. & Pain, C. C. 2024 Data assimilation with machine learning for dynamical systems: Modelling indoor ventilation. *Physica A* **643**, 129783.

Herzog, S., Wörgötter, F. & Parlitz, U. 2019 Convolutional autoencoder and conditional random fields hybrid for predicting spatial-temporal chaos. *Chaos* **29** (12), 123116.

Hewing, L., Wabersich, K. P., Menner, M. & Zeilinger, M. N. 2020 Learning-based model predictive control: Toward safe learning in control. *Annu. Rev. Control. Robotics Auton. Syst.:* **3**, 269–296.

Holmes, P., Lumley, J. L., Berkooz, G. & Rowley, C. W. 2012 Turbulence, Coherent Structures, Dynamical Systems and Symmetry, 2nd edn. *Cambridge Monographs on Mechanics* 2. Cambridge University Press.

Hotelling, H. 1992 Relations between two sets of variates. In *Breakthroughs in Statistics: methodology and distribution*, pp. 162–190. Springer.

Hou, Z.-S. & Wang, Z. 2013 From model-based control to data-driven control: Survey, classification and perspective. *Inf. Sci.* **235**, 3–35.

Hovland, S., Willcox, K. & Gravdahl, J. T. 2006 MPC for Large-Scale Systems via Model Reduction and Multiparametric Quadratic Programming. In *Proc. 45th IEEE Conf. Decis. Control*, pp. 3418–3423.

Howard, R. A. 1960 Dynamic Programming and Markov Processes. Oxford, England: John Wiley & Sons.

Hrovat, D., Di Cairano, S., Tseng, H. E. & Kolmanovsky, I. V. 2012 The development of model predictive control in automotive industry: A survey. In *2012 IEEE Int. Conf. Control Appl.*, pp. 295–302. IEEE.

Hussain, A. K. M. F. 1986 Coherent structures and turbulence. *J. Fluid Mech.* **173**, 303–356.

Hyman, J. M. & Nicolaenko, B. 1986 The kuramoto-sivashinsky equation: A bridge between PDE'S and dynamical systems. *Physica D* **18** (1), 113–126.

International Air Transport Association (IATA) 2025 Fly net zero: Decarbonizing aviation. Accessed: 12-Dec-2025.

Jiang, J., Wan, D. & Zhang, M. 2025 Numerical solutions of fixed points in two-dimensional kuramoto–sivashinsky equation expedited by reinforcement learning. *Proc. R. Soc. A* **481** (2306), 20240222.

Jiménez, J. 2013 Near-wall turbulence. *Phys. Fluids* **25** (10), 101302.

Juang, J.-N. & Pappa, R. S. 1985 An eigensystem realization algorithm for modal parameter identification and model reduction. *J. Guid. Control. Dyn.* **8** (5), 620–627.

Juang, J.-N., Phan, M., Horta, L. G. & Longman, R. W. 1993 Identification of observer/Kalman filter Markov parameters-Theory and experiments. *J. Guid. Control. Dyn.* **16** (2), 320–329.

Kaiser, E., Kutz, J. N. & Brunton, S. L. 2018 Sparse identification of nonlinear dynamics for model predictive control in the low-data limit. *Proc. R. Soc. A* **474** (2219), 20180335.

Kalman, R. E. 1960 A new approach to linear filtering and prediction problems. *J. Basic Eng.* **82** (1), 35–45.

Kalogirou, A., Keaveny, E. E. & Papageorgiou, D. T. 2015 An in-depth numerical study of the two-dimensional Kuramoto–Sivashinsky equation. *Proc. R. Soc. A* **471** (2179), 20140932.

Kang, W. & Fridman, E. 2021 Sampled-data control of 2-d kuramoto–sivashinsky equation. *IEEE Trans. Automat. Contr.* **67** (3), 1314–1326.

Karg, B. & Lucia, S. 2021 Approximate moving horizon estimation and robust nonlinear model predictive control via deep learning. *Comput. Chem. Eng.* **148**, 107266.

Kim, J. & Bewley, T. R. 2007 A linear systems approach to flow control. *Annu. Rev. Fluid Mech.* **39**, 383–417.

Kirk, D. E. 2004 Optimal control theory: an introduction. Courier Corporation.

Koller, T., Berkenkamp, F., Turchetta, M. & Krause, A. 2018 Learning-Based Model Predictive Control for Safe Exploration. In *2018 IEEE Conf. Decis. Control*, pp. 6059–6066.

Kramer, B., Peherstorfer, B. & Willcox, K. E. 2024 Learning nonlinear reduced models from data with operator inference. *Annu. Rev. Fluid Mech.* **56** (1), 521–548.

Krishna, K., Song, Z. & Brunton, S. L. 2022 Finite-horizon, energy-efficient trajectories in unsteady flows. *Proc. R. Soc. A* **478** (2258), 20210255.

Kuramoto, Y. 1978 Diffusion-induced chaos in reaction systems. *Prog. Theor. Phys. Suppl.* **64**, 346–367.

Kuramoto, Y. & Tsuzuki, T. 1976 Persistent propagation of concentration waves in dissipative media far from thermal equilibrium. *Prog. Theor. Phys.* **55** (2), 356–369.

Kurniawati, H. 2022 Partially Observable Markov Decision Processes and Robotics. *Annu. Rev. Control., Robot., Autom. Syst.* **5**, 253–277.

Kwok, J.-Y. & Tsang, I.-H. 2004 The pre-image problem in kernel methods. *IEEE Trans. Neural. Netw.* **15** (6), 1517–1525.

Lassila, T., Manzoni, A., Quarteroni, A. & Rozza, G. 2014 Model Order Reduction in Fluid Dynamics: Challenges and Perspectives, pp. 235–273. Cham: Springer International Publishing.

Launder, B. E. & Spalding, D. B. 1983 The numerical computation of turbulent flows. In *Numerical prediction of flow, heat transfer, turbulence and combustion*, pp. 96–116. Elsevier.

Lee, J. H. & Ricker, N. L. 1994 Extended Kalman filter based nonlinear model predictive control. *Ind. Eng. Chem. Res.* **33** (6), 1530–1541.

Li, S., Li, W. & Noack, B. R. 2022*a* Machine-learned control-oriented flow estimation for multi-actuator multi-sensor systems exemplified for the fluidic pinball. *J. Fluid Mech.* **952**, A36.

Li, Y., Cui, W., Jia, Q., Li, Q., Yang, Z., Morzyński, M. & Noack, B. R. 2022*b* Explorative gradient method for active drag reduction of the fluidic pinball and slanted ahmed body. *J. Fluid Mech.* **932**, A7.

Lin, B., He, X. & Ye, J. 2015 A geometric viewpoint of manifold learning. *Applied Informatics* **2** (1), 3.

Lin, F. 2007 Robust control design: an optimal control approach. John Wiley & Sons, Ltd.

Linot, A. J. & Graham, M. D. 2022 Data-driven reduced-order modeling of spatiotemporal chaos with neural ordinary differential equations. *Chaos* **32** (7), 073110.

Ljung, L. 2003 Convergence analysis of parametric identification methods. *IEEE Trans. Autom. Control.* **23** (5), 770–783.

Loiseau, J.-C. & Brunton, S. L. 2018 Constrained sparse galerkin regression. *J. Fluid Mech.* **838**, 42–67.

Lu, Q. & Zavala, V. M. 2021 Image-based model predictive control via dynamic mode decomposition. *J. Process Contr.* **104**, 146–157.

Lumley, J. L. 1967 The structure of inhomogeneous turbulent flows. *Atmos. Turbul. Radio Wave Propag.* pp. 166–178.

Luo, F.-M., Xu, T., Lai, H., Chen, X.-H., Zhang, W. & Yu, Y. 2024*a* A survey on model-based reinforcement learning. *Sci. China Inf. Sci.* **67** (2), 121101.

Luo, F.-M., Xu, T., Lai, H., Chen, X.-H., Zhang, W. & Yu, Y. 2024*b* A survey on model-based reinforcement learning. *Sci. China Inf. Sci.* **67** (2), 121101.

Lusch, B., Kutz, J. N. & Brunton, S. L. 2018 Deep learning for universal linear embeddings of nonlinear dynamics. *Nat. Commun.* **9** (1), 4950.

Macki, J. & Strauss, A. 1982 Introduction to Optimal Control Theory. *Undergraduate Texts in Mathematics* 1. New York, NY, USA: Springer.

Madhyastha, M., Li, G., Strnadová-Neeley, V., Browne, J., Vogelstein, J. T., Burns, R. & Priebe, C. E. 2020 Geodesic Forests. In *Proceedings of the 26th ACM SIGKDD Int. Conf. on Knowledge Discovery & Data Mining*, p. 513–523. New York, NY, USA: Association for Computing Machinery.

Marra, L., Meilán-Vila, A. & Discetti, S. 2024 Self-tuning model predictive control for wake flows. *J. Fluid Mech.* **983**, A26.

Maulik, R., Rao, V., Wang, J., Mengaldo, G., Constantinescu, E., Lusch, B., Balaprakash, P., Foster, I. & Kotamarthi, R. 2022 Efficient high-dimensional variational data assimilation with machine-learned reduced-order models. *Geosci. Model Dev.* **15** (8), 3433–3445.

Mayne, D. Q., Rawlings, J. B., Rao, C. V. & Scokaert, P. O. 2000 Constrained model predictive control: Stability and optimality. *Automatica* **36** (6), 789–814.

McCullough, D. 2015 The Wright Brothers. Simon and Schuster.

Mekuz, N. & Tsotsos, J. K. 2006 Parameterless Isomap with Adaptive Neighborhood Selection. In *Pattern Recognition*, pp. 364–373. Berlin, Heidelberg: Springer Berlin Heidelberg.

Mendez, M. A. 2023 Linear and nonlinear dimensionality reduction from fluid mechanics to machine learning. *Meas. Sci. Technol.* **34** (4), 042001.

Mentzelopoulos, A. P., Fan, D., Sapsi, T. P. & Triantafyllou, M. S. 2024 Variational autoencoders and transformers for multivariate time-series generative modeling and forecasting: Applications to vortex-induced vibrations. *Ocean Engineering* **310**, 118639.

Mesbah, A. 2018 Stochastic model predictive control with active uncertainty learning: A survey on dual control. *Annu. Rev. Control* **45**, 107–117.

MO, Y., TRAVERSO, T. & MAGRI, L. 2024 Decoder decomposition for the analysis of the latent space of nonlinear autoencoders with wind-tunnel experimental data. *DCE* **5**, e38.

MOERLAND, T. M., BROEKENS, J., PLAAT, A. & JONKER, C. M. 2023 Model-based reinforcement learning: A survey. *Found. Trends Mach. Learn.* **16** (1), 1–118.

MONTALÀ, R., FONT, B., SUÁREZ, P., RABAULT, J., LEHMKUHL, O., VINUESA, R. & RODRIGUEZ, I. 2025 Deep reinforcement learning for active flow control around a three-dimensional flow-separated wing at Re = 1,000. *arXiv preprint* .

MORENO-MORA, F., BECKENBACH, L. & STREIF, S. 2023 Predictive control with learning-based terminal costs using approximate value iteration. *IFAC-PapersOnLine* **56** (2), 3874–3879, 22nd IFAC World Congress.

MORTON, J., WITHERDEN, F. D., JAMESON, A. & KOCHENDERFER, M. J. 2018 Deep dynamical modeling and control of unsteady fluid flows. In *Proc. Int. Conf. Neural Inf. Process. Syst., NIPS '18*, p. 9278–9288. Red Hook, NY, USA: Curran Associates Inc.

NAKAMURA, T., FUKAMI, K., HASEGAWA, K., NABAE, Y. & FUKAGATA, K. 2021 Convolutional neural network and long short-term memory based reduced order surrogate for minimal turbulent channel flow. *Phys. Fluids* **33** (2), 025116.

NEPOMNYASHCHII, A. A. 1974 Stability of wavy conditions in a film flowing down an inclined plane. *Fluid. Dyn.* **9** (3), 354–359.

NEUMANN–BROSIG, M., MARCO, A., SCHWARZMANN, D. & TRIMPE, S. 2020 Data-efficient autotuning with bayesian optimization: An industrial control study. *IEEE Trans. Control Syst. Technol.* **28** (3), 730–740.

NIU, H., MCCALLUM, G. B., CHANG, A. B., KHAN, K. & AZAM, S. 2025 Exploring unsupervised feature extraction algorithms: tackling high dimensionality in small datasets. *Sci. Rep.* **15** (1), 21973.

NOACK, B. R. & MORZYŃSKI, M. 2017 The fluidic pinball—a toolkit for multiple-input multiple-output flow control (version 1.0). Chair of Virtual Engineering, Poznan University of Technology, Poland.

NÓVOA, A. & MAGRI, L. 2025 Online Model Learning with Data-Assimilated Reservoir Computers. In *Comput. Sci – ICCS 2025*, pp. 57–64. Cham: Springer Nature Switzerland.

NÓVOA, A., RACCA, A. & MAGRI, L. 2024 Inferring unknown unknowns: Regularized bias-aware ensemble Kalman filter. *Comput. Methods Appl. Mech. Eng.* **418**, 116502.

OLDEWURTEL, F., PARISIO, A., JONES, C. N., GYALISTRAS, D., GWERDER, M., STAUCH, V., LEHMANN, B. & MORARI, M. 2012 Use of model predictive control and weather forecasts for energy efficient building climate control. *Energy Build.* **45**, 15–27.

ÖZALP, E., NÓVOA, A. & MAGRI, L. 2026 Real-time forecasting of chaotic dynamics from sparse data and autoencoders. *Comput. Methods Appl. Mech. Eng.* **450**, 118600.

OZAN, D. E., NÓVOA, A. & MAGRI, L. 2025 Data-Assimilated Model-Based Reinforcement Learning for Partially Observed Chaotic Flows. In *Comput. Sci. – ICCS 2025 Workshops*, pp. 65–72. Cham: Springer Nature Switzerland.

PAGE, J., BRENNER, M. P. & KERSWELL, R. R. 2021 Revealing the state space of turbulence using machine learning. *Phys. Rev. Fluids* **6** (3), 034402.

PAPAGEORGIOU, D. T., MALDARELLI, C. & RUMSCHITZKI, D. S. 1990 Nonlinear interfacial stability of core-annular film flows. *Phys. Fluids* **2** (3), 340–352.

PAPAGEORGIOU, D. T. & SMYRLIS, Y. S. 1991 The route to chaos for the kuramoto-sivashinsky equation. *Theor. Comput. Fluid Dyn* **3** (1), 15–42.

PEARSON, K. 1901 On lines and planes of closest fit to systems of points in space. *Lond. Edinb. Dubl. Phil. Mag.* **2** (11), 559–572.

PEHERSTORFER, B. & WILLCOX, K. 2016 Data-driven operator inference for nonintrusive projection-based model reduction. *Comput. Methods Appl. Mech. Eng.* **306**, 196–215.

Peyron, M., Fillion, A., Gürol, S., Marchais, V., Gratton, S., Boudier, P. & Goret, G. 2021 Latent space data assimilation by using deep learning. *Q. J. R. Meteorol. Soc.* **147** (740), 3759–3777.

Polydoros, A. S. & Nalpantidis, L. 2017 Survey of model-based reinforcement learning: Applications on robotics. *J. Intell. Robot. Syst.* **86** (2), 153–173.

Pope, S. B. 2000 Turbulent Flows. Cambridge University Press.

Powell, W. B. 2007 Approximate Dynamic Programming: Solving the curses of dimensionality, *Wiley Series in Probability and Statistics*, vol. 703. John Wiley & Sons.

Prandtl, L. 1904 Ober Flüssigkeitsbewegung bei sehr kleiner Reibung. In *Proc. 3rd Int. Math. Congr.*, pp. 484–491. Heidelberg, Germany.

Propoi, A. 1963 Use of linear programming methods for synthesizing sampled-data automatic systems. *Automn. Remote Control* **24** (7), 837–844.

Protas, B. 2004 Linear feedback stabilization of laminar vortex shedding based on a point vortex model. *Phys. Fluids* **16** (12), 4473–4488.

Quade, M., Abel, M., Kutz, J. N. & Brunton, S. L. 2018 Sparse identification of nonlinear dynamics for rapid model recovery. *Chaos* **28** (6), 063116.

Rabault, J., Kuchta, M., Jensen, A., Réglade, U. & Cerardi, N. 2019 Artificial neural networks trained through deep reinforcement learning discover control strategies for active flow control. *J. Fluid Mech.* **865**, 281–302.

Racca, A., Doan, N. A. K. & Magri, L. 2023 Predicting turbulent dynamics with the convolutional autoencoder echo state network. *J. Fluid Mech.* **975**, A2.

Rahideh, A. & Shaheed, M. 2012 Constrained output feedback model predictive control for nonlinear systems. *Control Eng. Pract.* **20** (4), 431–443, special Section: IFAC Symposium on Advanced Control of Chemical Processes - ADCHEM 2009.

Raibaudo, C. & Martinuzzi, R. J. 2021 Unsteady actuation and feedback control of the experimental fluidic pinball using genetic programming. *Exp. Fluids* **62** (10), 219.

Raibaudo, C., Zhong, P., Noack, B. R. & Martinuzzi, R. J. 2020 Machine learning strategies applied to the control of a fluidic pinball. *Phys. Fluids* **32** (1), 015108.

Raissi, M., Perdikaris, P. & Karniadakis, G. E. 2019 Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **378**, 686–707.

Rawlings, J. B., Mayne, D. Q. & Diehl, M. 2020 Model predictive control: theory, computation, and design, 2nd edn. Nob Hill Publishing Madison, WI.

Reynolds, O. 1883 An experimental investigation of the circumstances which determine whether the motion of water shall be direct or sinuous, and of the law of resistance in parallel channels. *Philosophical Transactions of the Royal Society of London* (174), 935–982.

Richalet, J., Rault, A., Testud, J. & Papon, J. 1978 Model predictive heuristic control. *Automatica* **14** (5), 413–428.

Robinson, S. 1991 Coherent motions in the turbulent boundary layer. *Annu. Rev. Fluid Mech.* **23** (1), 601–639.

Rocha, P. R. B., de Sousa Almeida, J. L., de Paula Gomes, M. S. & Junior, A. C. N. 2023 Reduced-order modeling of the two-dimensional Rayleigh–Bénard convection flow through a non-intrusive operator inference. *Eng. Appl. Artif. Intell.* **126**, 106923.

Roussopoulos, K. 1993 Feedback control of vortex shedding at low Reynolds numbers. *J. Fluid Mech.* **248**, 267–296.

Rowley, C. W. 2005 Model reduction for fluids, using balanced proper orthogonal decomposition. *Int. J. Bifurcat. Chaos* **15** (03), 997–1013.

Rowley, C. W. & Williams, D. R. 2006 Dynamics and control of high-Reynolds-number flow over open cavities. *Annu. Rev. Fluid Mech.* **38**, 251–276.

Saltik, M. B., Özkan, L., Ludlage, J. H. A., Weiland, S. & Van den Hof, P. M. J. 2018 An outlook on robust model predictive control algorithms: Reflections on performance and computational aspects. *J. Process Control* **61**, 77–102.

Sangiorgio, M., Dercole, F. & Guariso, G. 2021 Forecasting of noisy chaotic systems with deep neural networks. *Chaos, Solitons & Fract.* **153**, 111570.

Schmid, P. J. 2010 Dynamic mode decomposition of numerical and experimental data. *J. Fluid Mech.* **656**, 5–28.

Schölkopf, B. & Smola, A. J. 2002 Learning with kernels: support vector machines, regularization, optimization, and beyond. MIT Press, Cambridge, MA.

Semeraro, O. 2025 Reinforcement Learning for Fluid Mechanics: an overview on Fundamentals from a Control Perspective. In *Machine Learning for Fluid Dynamics*.

Sethi, S. P. 2021 What Is Optimal Control Theory?, pp. 1–23. Cham: Springer.

Shi, Y. & Zhang, K. 2021 Advanced model predictive control framework for autonomous intelligent mechatronic systems: A tutorial overview and perspectives. *Annu. Rev. Control* **52**, 170–196.

Sipp, D., Marquet, O., Meliga, P. & Barbagallo, A. 2010 Dynamics and control of global instabilities in open-flows: A linearized approach. *Appl. Mech. Rev.* **63** (3), 030801.

Siv Ashinsky, G. I. 1988 Nonlinear analysis of hydrodynamic instability in laminar flames—I. Derivation of basic equations. In *Dynamics of Curved Fronts*, pp. 459–488. San Diego: Academic Press.

Sivashinsky, G. I. 1980 On flame propagation under conditions of stoichiometry. *SIAM J. Appl. Math.* **39** (1), 67–82.

Smagorinsky, J. 1963 General circulation experiments with the primitive equations: I. The basic experiment. *Mon. Weather Rev.* **91** (3), 99–164.

Solera-Rico, A., Sanmiguel Vila, C. & Discetti, S. 2025 A framework for realisable data-driven active flow control using model predictive control applied to a simplified truck wake. *arXiv preprint* .

Solera-Rico, A., Sanmiguel Vila, C., Gómez-López, M., Wang, Y., Almashjary, A., Dawson, S. T. M. & Vinuesa, R. 2024 β-variational autoencoders and transformers for reduced-order modelling of fluid flows. *Nat. Commun.* **15** (1), 1361.

Souvenir, R. & Pless, R. 2005 Isomap and nonparametric models of image deformation. *Seventh IEEE WACV/MOTION 2005* **2**, 195–200.

Suárez, P., Alcántara-Ávila, F., Rabault, J., Miró, A., Font, B., Lehmkuhl, O. & Vinuesa, R. 2025 Flow control of three-dimensional cylinders transitioning to turbulence via multi-agent reinforcement learning. *Comms. Eng.* **4** (1), 113.

Suetani, H., Soejima, K., Matsuoka, R., Parlitz, U. & Hata, H. 2012 Manifold learning approach for chaos in the dripping faucet. *Phys. Rev. E* **86**, 036209.

Sutton, R. S. & Barto, A. G. 2018 Reinforcement Learning: An Introduction, 2nd edn. Cambridge, MA: MIT Press.

Tauro, F., Grimaldi, S. & Porfiri, M. 2014 Unraveling flow patterns through nonlinear manifold learning. *PLoS One* **9** (3), e91131.

Tenenbaum, J. B., Silva, V. D. & Langford, J. C. 2000 A global geometric framework for nonlinear dimensionality reduction. *Science* **290** (5500), 2319–2323.

Tibshirani, R. 1996 Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **58** (1), 267–288.

Tolle, K. M., Tansley, D. S. W. & Hey, A. J. G. 2011 The fourth paradigm: data-intensive scientific discovery [point of view]. *Proc. IEEE* **99** (8), 1334–1337.

Tzeng, J., Lu, H. H.-S. & Li, W.-H. 2008 Multidimensional scaling for large genomic data sets. *BMC Bioinform.* **9** (1), 179.

URIBARRI, G. & MINDLIN, G. B. 2020 The structure of reconstructed flows in latent spaces. *Chaos* **30** (9), 093109.

UY, W. I. T., WANG, Y., WEN, Y. & PEHERSTORFER, B. 2023 Active operator inference for learning low-dimensional dynamical-system models from noisy data. *SIAM J. Sci. Comput.* **45** (4), A1462–A1490.

VAZQUEZ, S., LEON, J. I., FRANQUELO, L. G., RODRIGUEZ, J., YOUNG, H. A., MARQUEZ, A. & ZANCHETTA, P. 2014 Model predictive control: A review of its applications in power electronics. *IEEE Ind. Electron. Mag.* **8** (1), 16–31.

VIGNON, C., RABAULT, J. & VINUESA, R. 2023 Recent advances in applying deep reinforcement learning for flow control: Perspectives and future directions. *Phys. Fluids* **35** (3), 031301.

VLACHAS, P. R., ARAMPATZIS, G., UHLER, C. & KOUMOUTSAKOS, P. 2022 Multiscale simulations of complex systems by learning their effective dynamics. *Nat. Mach. Intell.* **4** (4), 359–366.

WABERSICH, K. P., HEWING, L., CARRON, A. & ZEILINGER, M. N. 2021 Probabilistic model predictive safety certification for learning-based control. *IEEE Trans. Autom. Control.* **67** (1), 176–188.

WALL, M. E., RECHTSTEINER, A. & ROCHA, L. M. 2003 Singular Value Decomposition and Principal Component Analysis, pp. 91–109. Boston, MA: Springer US.

WANG, Q., YAN, L., HU, G., CHEN, W., RABAULT, J. & NOACK, B. R. 2024 Dynamic feature-based deep reinforcement learning for flow control of circular cylinder with sparse surface pressure sensing. *J. Fluid Mech.* **988**, A4.

WANG, X., DENG, N., CORNEJO MACEDA, G. Y. & NOACK, B. R. 2023 Cluster-based control for net drag reduction of the fluidic pinball. *Phys. Fluids* **35** (2), 023601.

VAN DE WATER, H. & WILLEMS, J. 1981 The certainty equivalence property in stochastic control theory. *IEEE Trans. Autom. Control.* **26** (5), 1080–1087.

WEINER, A. & GEISE, J. 2025 Model-based deep reinforcement learning for accelerated learning from flow simulations. *Mech.* **60** (6), 1771–1788.

WEISSENBACHER, M., BOROVYKH, A. & RIGAS, G. 2025 Reinforcement learning of chaotic systems control in partially observable environments. *Flow Turbul. Combust.* pp. 1–22.

WILLIAMS, J. P., KUTZ, J. N. & MANOHAR, K. 2024 Reservoir computing for system identification and predictive control with limited data. *arXiv preprint* .

WITTENMARK, B. 1995 Adaptive dual control methods: An overview. *IFAC* **28** (13), 67–72.

XIA, C., ZHANG, J., KERRIGAN, E. C. & RIGAS, G. 2024 Active flow control for bluff body drag reduction using reinforcement learning with partial measurements. *J. Fluid Mech.* **981**, A17.

XIANG, J., ZONG, H., WU, Y., LI, J. & LIANG, H. 2024 Experimental control of the flow separation behind a backward facing step using deep reinforcement learning. *Phys. Fluids* **36** (10), 105102.

YE, M. & ELSHEIKH, A. H. 2025 Model-based reinforcement learning for active flow control. *Phys. Fluids* **37** (9), 093363.

YOUSAF, M., SHAKOOR KHAN, M. S. & ULLAH, S. 2024 An extended-isomap for high-dimensional data accuracy and efficiency: a comprehensive survey. *Multim. Tools Appl.* **83** (38), 85523–85574.

ZAFRA, E., VAZQUEZ, S., GEYER, T., AGUILERA, R. P. & FRANQUELO, L. G. 2023 Long prediction horizon fcs-mpc for power converters and drives. *IEEE open J. Ind. Electron. Soc.* **4**, 159–175.

ZAHED, M. & SKAFYAN, M. 2025 Silhouette-based evaluation of PCA, isomap, and t-SNE on linear and nonlinear data structures. *Stats* **8** (4), 105.

ZENG, J., ZHANG, B. & SREENATH, K. 2021 Safety-Critical Model Predictive Control with Discrete-Time Control Barrier Function. In *Proc. Amer. Control Conf.*, pp. 3882–3889.

Özalp, E., Margazoglou, G. & Magri, L. 2023 Reconstruction, forecasting, and stability of chaotic dynamics from partial data. *Chaos* **33** (9), 093107.

Part II

Journal Publications

Summary of the papers

Paper 1

Self-tuning model predictive control for wake flows

This work presents a self-tuning Model Predictive Control (MPC) framework for the real-time control of wakes with minimal user intervention. The approach addresses two main challenges of MPC in fluid flows: the strong dependence on hand-tuned hyperparameters and the sensitivity to measurement noise and model uncertainty. To overcome these limitations, the proposed architecture combines (i) data-driven identification of a compact plant model using the Sparse Identification of Nonlinear Dynamics (SINDy) to obtain, (ii) automated hyperparameter selection via Bayesian Optimisation (BO), and (iii) noise-robust state estimation based on Local Polynomial Regression (LPR). The SINDy modelling strategy is employed because it yields an interpretable dynamical model to be integrated into the control framework. The method is applied to the two-dimensional wake of the fluidic pinball, a configuration of three independently rotating cylinders exhibiting chaotic dynamics at $Re_D = 150$. The plant model relies exclusively on aerodynamic force measurements, drag and lift coefficients and their time derivatives, together with the actuation inputs. The MPC cost function accounts for drag reduction, lift fluctuations, actuation power and actuation smoothness, while respecting constraints on rotation rates. Closed-loop simulations demonstrate that BO systematically enhances controller performance compared to hand-tuned MPC, reducing drag and suppressing excessive lift oscillations. LPR enables robust force estimation even under significant measurement noise. The resulting actuation strategy consistently exploits boat-tailing for drag reduction and stagnation-point manipulation for lift stabilisation. The study establishes an automated and noise-tolerant MPC workflow that substantially reduces the need for manual tuning. While high-level design choices such as performance metrics and optimisation settings remain user-defined, the generality and reliance of the framework on limited sensing make it suitable for applications in flow control where plant models are uncertain or noisy.

Paper 2

Latent-space non-linear model predictive control for partially-observable systems

This work introduces a scalable control framework for high-dimensional and partially observable systems based on nonlinear Model Predictive Control (MPC). To overcome the challenges posed by large system size and partial observability, the framework combines data-driven model reduction, latent-space control, and state estimation into a unified formulation. Specifically, a predictive model is learned via Operator Inference on a reduced Proper Orthogonal Decomposition (POD) basis, yielding a compact latent representation that captures the dominant system dynamics. State estimation in the latent space is achieved using an Unscented Kalman Filter (UKF), allowing reconstruction from sparse and noisy measurements and enabling closed-loop operation under partial observation. The methodology is validated on one- and two-dimensional Kuramoto–Sivashinsky equation (KS), which serve as prototypical benchmarks for high-dimensional, spatially-extended, and chaotic systems. In numerical experiments, the latent-space MPC successfully stabilises the dynamics, even when only a limited number of sensors are available, and measurements are contaminated by noise. By computing control inputs directly in the reduced latent space, the method achieves significant computational efficiency without sacrificing predictive or control performance. While the current work relies on a specific latent-space encoding, the framework is general and can be extended to more complex systems by adopting more flexible or expressive encoders, such as autoencoders based on deep neural networks, which can capture richer nonlinear manifold structures. This approach demonstrates a practical and effective path for nonlinear control of high-dimensional systems where full-state measurement is impractical, and it highlights the potential of combining model reduction, estimation, and latent-space MPC for real-time control of large-scale spatio-temporal systems with limited sensing information.

Paper 3

Actuation manifold from snapshot data

This work presents a data-driven framework to identify a low-dimensional manifold that captures the dynamics of controlled fluid flows. The approach relies exclusively on snapshot data generated for a representative set of constant actuation inputs. The manifold is constructed using isometric mapping (ISOMAP) as an encoder to preserve intrinsic geometric structure, and a hybrid decoder combining a neural network with k-Nearest-Neighbour interpolation to reconstruct the full flow field. The methodology is assessed on the fluidic pinball at a Reynolds number of $Re_D = 30$, where the unforced wake exhibits statistically symmetric periodic vortex shedding, corresponding to a one-dimensional limit cycle. The actuation consists of independent constant rotations of the cylinders. The learned actuation manifold is five-dimensional and achieves a small reconstruction error across a wide range of actuation conditions. Remarkably, the manifold coordinates self-organise into physically interpretable directions: two coordinates represent the downstream periodic shedding, while the remaining three encode key near-field actuation mechanisms, boat-tailing, Magnus effect, and forward stagnation-point control. These coordinates emerge without imposing physical constraints, highlighting the method's ability to extract meaningful structure directly from data. The results demonstrate that the actuation manifold provides a compact and expressive representation of controlled wake dynamics, offering a promising foundation for flow estimation and future model-based control strategies.

1

Paper 1

Self-tuning model predictive control for wake flows

L. Marra[1], A. Meilán-Vila[2] & S. Discetti[1]

[1] Department of Aerospace Engineering, Universidad Carlos III de Madrid, Av. de la Universidad 30, Leganés, 28911, Madrid, Spain

[2] Department of Statistics, Universidad Carlos III de Madrid, Av. de la Universidad 30, Leganés, 28911, Madrid, Spain

Journal of Fluid Mechanics (2024), vol. **983**, A26. DOI: `https://doi.org/10.1017/jfm.2024.47`

This study presents a noise-robust closed-loop control strategy for wake flows employing model predictive control. The proposed control framework involves the autonomous offline selection of hyperparameters, eliminating the need for user interaction. To this purpose, Bayesian optimization maximises the control performance, adapting to external disturbances, plant model inaccuracies and actuation constraints. The noise robustness of the control is achieved through sensor data smoothing based on local polynomial regression. The plant model can be identified through either theoretical formulation or using existing data-driven techniques. In this work we leverage the latter approach, which requires minimal user intervention. The self-tuned control strategy is applied to the control of the wake of the fluidic pinball, with the plant model based solely on aerodynamic force measurements. The closed-loop actuation results in two distinct control mechanisms: boat tailing for drag reduction and stagnation point control for lift stabilization. The control strategy proves to be highly effective even in realistic noise scenarios, despite relying on a plant model based on a reduced number of sensors.

Key words: Machine learning, Wakes

---

1. Introduction

Prediction and control of fluid flows to pursue a specific objective is a highly compelling research area (Gad-el Hak, 2000). Flow control offers wide-ranging practical applications in diverse fields, including vehicle dynamics, aircraft and marine transportation, meteorology, energy production from water and wind, combustion and chemical processes, and more (Duriez *et al.*, 2017). The goals of fluid flow control generally encompass, among others, drag reduction, control of separation and transition, lift or mixing enhancement. In recent years, drag reduction has received considerable attention due to its notable impact on the environmental footprint of transportation means (Green, 2003).

In the last decades, active flow control has garnered increasing attention. This technique can be implemented in open-loop and closed-loop configurations. The former involves predetermining the actuation law irrespective of the system state, thus simplifying its application. Notable examples include wake control of bluff bodies (Blackburn & Henderson, 1999; Cetiner & Rockwell, 2001; Thiria *et al.*, 2006; Parkin *et al.*, 2014), open-cavity flows (Sipp, 2012; Little *et al.*, 2007; Nagarajan *et al.*, 2018) and heat transfer (Castellanos *et al.*, 2022*b*) among others. However, open-loop control's effectiveness is limited in unstable flow stabilization, responding to changes in the controlled system parameters or dealing with external disturbances. On the contrary, closed-loop implementations (also referred to as reactive control) involve feeding the control law by the knowledge of the state. This approach offers greater flexibility and adaptability (Brunton & Noack, 2015). Experimental evidence demonstrates the superior performance of closed-loop over open-loop control; see e.g. Pinier *et al.* (2007) or Shimomura *et al.* (2020).

The identification of control laws requires adequate knowledge of the system dynamics and its response to control inputs. In fluid dynamics, model-based techniques have traditionally been utilised to obtain this information, proving successful in various scenarios (Kim & Bewley, 2007). Examples of applications include transition delay in spatially evolving wall-bounded flows (Monokrousos *et al.*, 2008; Chevalier *et al.*, 2007; Tol *et al.*, 2019), cavity flow control (Rowley & Williams, 2006; Illingworth *et al.*, 2011), separation control on a low-Reynolds-number airfoil (Ahuja *et al.*, 2007), wake stabilization of cylinders (Schumm *et al.*, 1994; Gerhard *et al.*, 2003; Tadmor *et al.*, 2011), skin-friction drag reduction (Cortelezzi & Speyer, 1998; Lee *et al.*, 2001; Kim, 2011). However, the identification of efficient analytical control laws faces an important challenge in the presence of complex nonlinear multiscale dynamics.

In recent years, model-free techniques have gained popularity, driven by advancements in hardware and the increasing efficiency of data-driven and machine-learning algorithms. Examples of model-free techniques include genetic algorithms in jet mixing optimization (Koumoutsakos *et al.*, 2001; Wu *et al.*, 2018), wake flows (Poncet *et al.*, 2005; Raibaudo *et al.*, 2020), separation control (Gautier *et al.*, 2015) and combustion noise (Buche *et al.*, 2002). Reinforcement learning (RL) has also recently gained popularity, with successful applications in the control of bluff body wakes (Rabault *et al.*, 2019; Fan *et al.*, 2020; Castellanos *et al.*, 2022*a*) and natural convection (Beintema *et al.*, 2020). Despite the encouraging results of such model-free techniques, their effectiveness is limited by the need for large datasets.

Within model-based techniques, model predictive control (MPC) offers interesting features to deal with the challenges of fluid flow control. Model predictive control is based on the idea of receding horizon control. It has found application in the industry since the 1980s (Qin & Badgwell, 2003), in particular with extensive use in refineries and the petrochemical industry (Lee, 2011). Model predictive control has demonstrated excellent performance in controlling complex systems with constraints, strong nonlinearities, and time delays (Henson, 1998; Allgöwer *et al.*, 2004; Camacho & Alba, 2013; Grüne & Pannek, 2017). Therefore, it is particularly appropriate for complex systems that challenge traditional linear controllers (Corona & De Schutter, 2008). The method requires the identification of a model of the system dynamics capable of predicting its behaviour under exogenous inputs. The optimal control is determined through the iterative solution of an optimization problem within a prediction window, aiming to minimise a user-defined cost function that considers the distance of the system state from the control target. Moreover, model predictive control allows for the straightforward implementation of hard constraints, such as hardware limitations, distinguishing it from classical control approaches. Model predictive control has been successfully applied in the control of complex fluid systems, see, e.g. Collis *et al.* (2000); Bewley *et al.* (2001); Bieker *et al.* (2020); Sasaki & Tsubakino (2020); Morton *et al.* (2018); Peitz *et al.* (2020). A crucial aspect of MPC implementation is achieving a proper balance among the terms of the loss function. The user needs to select weights (referred to as hyperparameters) for the loss, considering factors like closeness to the target, cost of the action and other application-tailored constraints. This choice has a clear impact on the final performance. In flow control applications this process traditionally relies on user experience, which poses the risk of suboptimal choices.

Bayesian optimization (BO) or RL techniques have demonstrated excellent results in hyperparameter tuning, particularly in the fields of autonomous driving and robotics (Edwards *et al.*, 2021; Fröhlich *et al.*, 2022; Bøhn *et al.*, 2023). A comprehensive review in this area can be found in Hewing *et al.* (2020). However, in the application of nonlinear MPC to flow control, examples are scarce, and the choice of MPC parameters is often guided by trial error and intuition. This approach risks falling into suboptimal configurations that may not adequately account for the different degrees of fidelity in the terms involved in the loss function. This issue is particularly relevant in fluid mechanics, where the uncertainty in predicting the plant behaviour and the measured state/control actions should play a role in the parameter selection process. Unfortunately, an analytical formulation is elusive in most cases.

Moreover, as a closed-loop strategy, the implementation of MPC requires feedback, consisting of time-sampled measurements of a feature of the system to be controlled. In real control scenarios, this sampling is often affected by measurement noise, which can compromise control decision making. Thus, suitable smoothing techniques are necessary to enhance noise robustness. In time series analysis a non-parametric statistical technique called local polynomial regression (LPR) proves particularly effective in this task. Local polynomial regression estimates the regression function of sensor outputs and their time derivatives without assuming any prior information. Applications of LPR for control purposes are described in works such as Steffen *et al.* (2010) or Ouyang *et al.* (2018).

In this paper, we propose a fully automatic architecture that self-tunes control and optimization process parameters with minimal user input. Our MPC framework adapts to different levels of noise and/or limited state knowledge. The methodology builds upon offline black-box optimization via Bayesian methods for hyperparameter tuning. Furthermore, we discuss the robustness enhancement to noise using an online LPR. The effectiveness of the control algorithm is evaluated through its application to the control of the wake of the fluidic pinball (Deng *et al.*, 2020) in the chaotic regime. Although not strictly needed, plant identification is also data driven. In this work, nonlinear system identification is performed using the sparse identification of nonlinear dynamics with control (SINDYc, Brunton *et al.*, 2016*b*).

The paper is organised as follows. Section 2 provides a description of the methodology, emphasizing the mathematical tools and the MPC framework employed. Additionally, this section includes specific details regarding the chosen test case for illustration purposes. The results of the control application, along with their interpretation are provided in § 3. Finally, the conclusions are discussed in § 4.

2. Methodology

This section presents the backbone of the MPC algorithm, with a detailed description of the mathematical tools involved in it. Figure 1 includes a diagram illustrating all the required steps for its implementation. In addition, Algorithm 1 is introduced to give more details on the procedure.

The main block in the schematics represents the MPC algorithm, following the approach proposed by Kaiser *et al.* (2018). The main novelty in this module is the robustness enhancement by online filtering with LPR. This is particularly useful when the plant dynamics is modelled with time-delay coordinates or their derivatives. Local polynomial regression is directly applied to past sensor data for online implementation.

The roadmap suggests several necessary steps before implementing the control. First, a training dataset is generated. The dataset consists of the time series of the state dynamics under different exogenous inputs. This data can be collected using various methods, including simulations or experiments. The system state and exogenous inputs should be defined based on the specific system being controlled. Second, a plant model is defined to predict the system behaviour. In this work we use a data-driven nonlinear system identification. The final step, before the implementation of the control, focuses on tuning the parameters that define the MPC cost function. Control performance is significantly influenced by their selection. The self-tuning of the hyperparameters is the core of our framework. This tuning is carried out using a BO algorithm.

An important aspect of the proposed method is the need for minimal user interaction. Indeed, the two main inputs are the reference set point (i.e. the control target) and a cost function for the MPC algorithm. The weight of the different contributions in the cost function will be determined in the MPC tuning. Bayesian optimization automatically adjusts to different levels of noise and uncertainties. This greatly enhances the usability and adaptability of the framework to different systems.

2.1. *Self-tuning model predictive control*

2.1.1. *Model predictive control*

This section describes the MPC implementation. It is assumed to have a time-evolving process whose complete description relies on $N_a \geq 1$ parameters. These are included in a state vector, denoted at a given instant t as $\boldsymbol{a} \equiv \boldsymbol{a}(t) = \big(a^1(t), \ldots, a^{N_a}(t)\big)'$. The evolution in time of the process can be influenced by the choice of $N_b \geq 1$ exogenous parameters. These are included in an input vector $\boldsymbol{b} \equiv \boldsymbol{b}(t) = \big(b^1(t), \ldots, b^{N_b}(t)\big)'$, $\boldsymbol{b} \in \mathcal{B} \subset \mathbb{R}^{N_b}$, where $\mathcal{B}$ is the set of allowable inputs. Denoting the time derivative of the state vector with $\dot{\boldsymbol{a}}$, the system dynamics is described by the following set of equations:

$$\begin{array}{rcl} \dot{\boldsymbol{a}} & = & f(\boldsymbol{a}, \boldsymbol{b}) \\ \boldsymbol{a}(0) & = & \boldsymbol{a}_0. \end{array} \tag{1}$$

Here $\boldsymbol{a}(t_j) = \boldsymbol{a}_j$ and f is the function characterizing the system's temporal evolution. The process is considered as controlled. This means that for each time t, the input vector $\boldsymbol{b}(t)$ can be selected in order to manipulate the system dynamics according to a specific objective. More specifically, the aim is to control $N_c \geq 1$ features of the dynamical system, included in the vector $\boldsymbol{c} \equiv \boldsymbol{c}(t) = \big(c^1(t), \ldots, c^{N_c}(t)\big)'$. Note that the vector $\boldsymbol{c}$ is dependent on the system state. In this context, it is assumed that the target features are part of the state vector itself, although this may not always be applicable. The objective is to achieve control over the vector $\boldsymbol{c}$ by ensuring that it closely tends to a desired reference $\boldsymbol{c}_* \in \mathbb{R}^{N_c}$ over time. Model predictive control can be used for set-point stabilization, trajectory tracking or path following (see Raković & Levine, 2018, pp. 169-198), depending on the choice of $\boldsymbol{c}_*$.

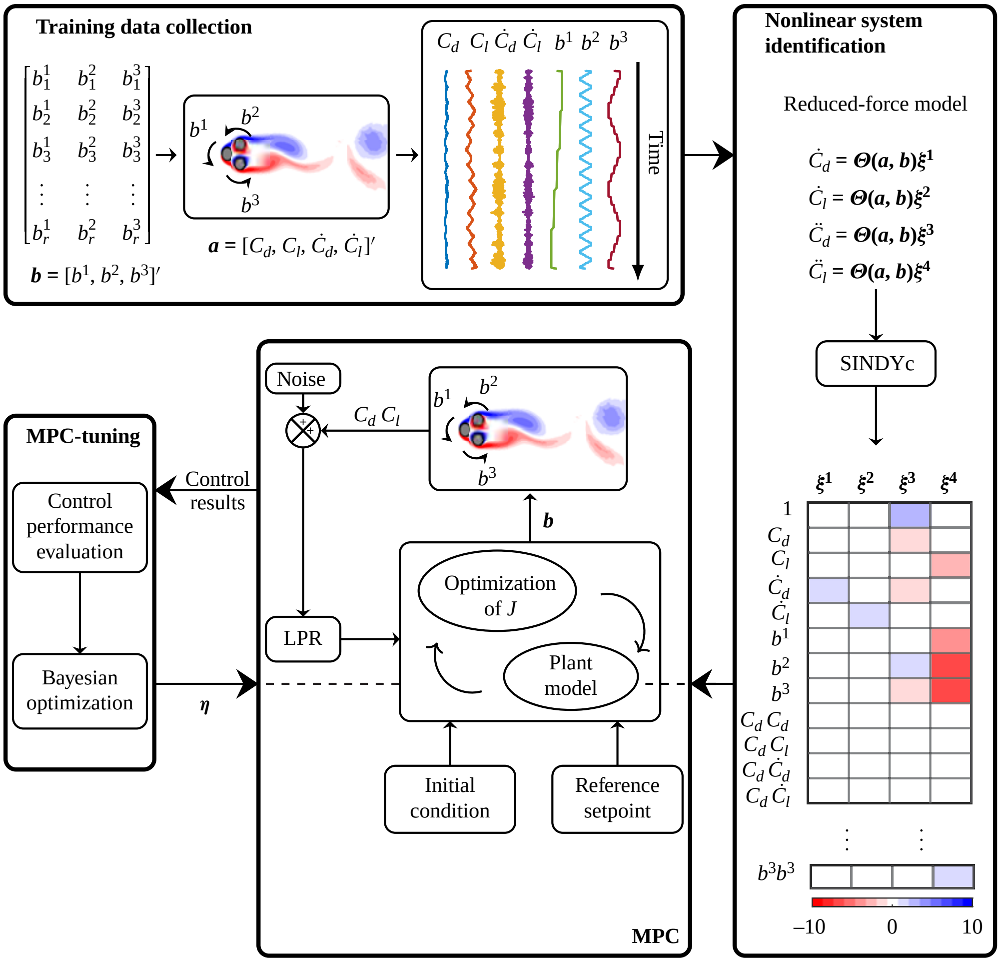

Figure 1: The MPC algorithm schematic: dataset generation, creation of the predictive model and parameter tuning. The main block displays the closed-loop MPC scheme, including also LPR to mitigate the effects of sensor measurement noise.

For the purpose of a control application, $\boldsymbol{a}$, $\boldsymbol{b}$ and $\boldsymbol{c}$ are sampled over a discrete-time vector, equispaced with a fixed time interval T_s. This discrete-time representation is essential to determine how often the exogenous input is updated. The input is assumed to be constant between consecutive time steps of the control. The implementation of the MPC follows a series of sequential procedures, as can be seen in Algorithm 1. An illustration of the process is provided in Figure 2.

Firstly, the control process starts from a time instant t_j, where a measurement $\boldsymbol{s}_j$ of the state vector is available. It is assumed that the entire vector of target features is observed. A conditional prediction of the state vector is obtained by a model of the dynamics. This prediction under a given input sequence, referred to as $\hat{\boldsymbol{a}}_{j+k|j}$, is generated

Algorithm 1 Control algorithm

Training data collection

Ensure: Training dataset of fluidic pinball forces

```
Choose open loop actuations
Resolve fluidic pinball wake using DNS
Post process flow fields for C_d and C_l computation
```

Nonlinear system identification (§ 2.1.2)

Ensure: SINDYc fluidic pinball force model: $\dot{a}^k = \boldsymbol{\Theta}(\boldsymbol{a}, \boldsymbol{b})\hat{\boldsymbol{\xi}}^k, \quad k = 1, \ldots, N_a$ in (9)

Construct the matrices $\mathbf{A}, \dot{\mathbf{A}}$ and $\mathbf{B}$ in (5) using training dataset

Compute SINDYc active terms of the dynamics:

$\hat{\boldsymbol{\xi}}^k = arg \underset{\boldsymbol{\xi^k}}{min} \left\{ \frac{1}{2} \left\| \dot{\mathbf{A}}_{\bullet,k} - \boldsymbol{\Theta}(\mathbf{A}, \mathbf{B})\boldsymbol{\xi}^k \right\|_2^2 + \lambda \left\| \boldsymbol{\xi}^k \right\|_1 \right\}$ in (8)

MPC-tuning (§ 2.1.4)

Ensure: Optimal hyperparameter vector $\boldsymbol{\eta}_{opt}$ solution of (19)

Select different vectors of hyperparameters for seeding

while BO stopping criterion not met **do**

Update $\mathcal{J}_{BO}$ sampling points

Use GP to update posterior distribution of $\mathcal{J}_{BO}$ given the samples available so far

Build and optimise the acquisition function to find the next search point

end while

MPC application (§ 2.1.1)

for $j = 1, \ldots, n_{BO}$ **do**

Measure system state

  if there is noise in the measure **then**

    Use LPR to last sensor data (§ 2.1.3)

  end if

Optimization of MPC cost function in (2) using hyperparameters optimised in MPC-tuning

Apply first optimal control component $\boldsymbol{b}_{j+1} = \boldsymbol{b}_1^{opt}$

Advance fluidic pinball wake of T_s time units

end for

within a prediction window $t_{j+k}, k = 1, \ldots, w_p$. Consequently, a prediction of the target features vector $\hat{\boldsymbol{c}}_{j+k|j}$ is obtained.

The optimal input sequence $\{\boldsymbol{b}_k^{opt}\}_{k=1}^{w_c}$ can be determined in a control window $t_{j+k}, k = 1, \ldots, w_c$, by minimizing a cost function $\mathcal{J}_{MPC} : \mathbb{R}^{N_b} \to \mathbb{R}^+$. A common choice for the cost function is

$$\begin{aligned} \mathcal{J}_{MPC}(\boldsymbol{b}) \;\; = \;\; & \textstyle\sum_{k=0}^{w_p} \left\| \hat{\boldsymbol{c}}_{j+k|j} - \boldsymbol{c}_* \right\|_{\mathbf{Q}}^2 + \\ & + \textstyle\sum_{k=1}^{w_c} (\left\| \boldsymbol{b}_{j+k|j} \right\|_{\mathbf{R}_b}^2 + \left\| \Delta \boldsymbol{b}_{j+k|j} \right\|_{\mathbf{R}_{\Delta b}}^2), \end{aligned} \tag{2}$$

where $\mathbf{Q} \in \mathbb{R}^{N_c \times N_c}$ and $\mathbf{R}_b, \mathbf{R}_{\Delta b} \in \mathbb{R}^{N_b \times N_b}$ are positive and semi-positive definite weight matrices, respectively. In addition, $\|\boldsymbol{d}\|_{\mathbf{M}}^2 = \boldsymbol{d}'\mathbf{M}\boldsymbol{d}$ represents the weighted norm of a

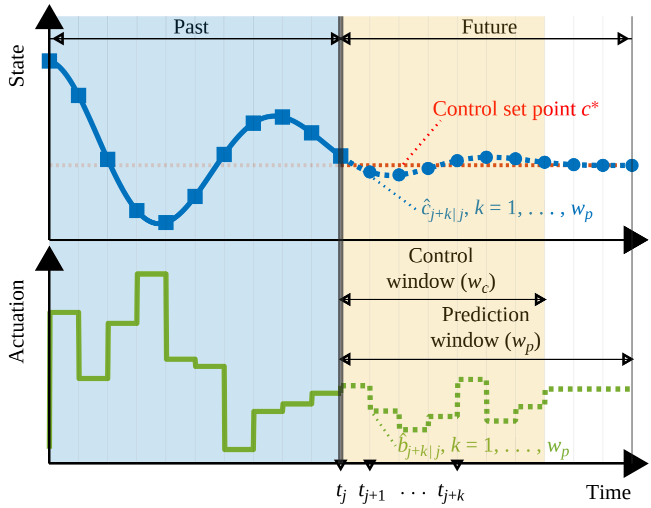

Figure 2: Graphical representation of MPC strategy for stabilizing around a set point (horizontal dashed line). Past measurements (light-blue-shaded region) depict system state (blue lines with squares) and actuation (green line). The control window w_c is shown in orange. Dashed lines indicate future state and actuation predictions. Blue circles represent a discrete sampling of the system state prediction. The continuous formulation allows non-mandatory discrete sampling and step-like actuation can be relaxed.

generic vector $\boldsymbol{d}$ with respect to a symmetric and positive definite matrix $\mathbf{M}$, where $\boldsymbol{d}'$ denotes the transposition of the vector $\boldsymbol{d}$. The variable $\Delta \boldsymbol{b}_{j+k|j}$ denotes the input variability in time, that is, $\boldsymbol{b}_{j+k|j} - \boldsymbol{b}_{j+k-1|j}$. In the definition of the cost function, the errors in state predictions with respect to the reference trajectory, i.e. $\hat{\boldsymbol{c}}_{j+k|j} - \boldsymbol{c}_*$, are penalised, as well as the actuation cost and variability. In this paper the aforementioned weight matrices are assumed to be diagonal; thus:

$$\begin{aligned} \mathbf{Q} &= \operatorname{diag}(Q^1, \ldots, Q^{N_c}), \\ \mathbf{R}_b &= \operatorname{diag}(R_{b^1}, \ldots, R_{b^{N_b}}), \\ \mathbf{R}_{\Delta b} &= \operatorname{diag}(R_{\Delta b^1}, \ldots, R_{\Delta b^{N_b}}). \end{aligned} \tag{3}$$

Once the optimization problem in (2) is solved, only the first component of the optimal control sequence, $\boldsymbol{b}_{j+1} = \boldsymbol{b}_1^{opt}$, is applied. The optimization is then reinitialised and repeated at each subsequent time step of the control. Generally, the prediction window covers a wider range than the control window ($w_p \geq w_c$). The control vector is considered constant beyond the end of the control window, as discussed in Kaiser *et al.* (2018).

Hard constraints on the input vector can readily be incorporated into the optimization process. At each time step, the optimal problem must guarantee that $\boldsymbol{b} \in \mathcal{B}$, where $\mathcal{B}$ is generally determined by the control hardware. The set of allowable inputs is

$$\begin{aligned} b_j^k &\in [b_{min}^k\ , \ b_{max}^k], \\ \Delta b_j^k &\in [\Delta b_{min}^k, \Delta b_{max}^k], \quad j = 1, \ldots, t \ \text{ and } \ k = 1, \ldots, N_b, \end{aligned} \tag{4}$$

where b_j^k and Δb_j^k are the k^{th} control input and input variability component at the j^{th} time step. The superscripts *min* and *max* indicate the minimum and maximum admissible values for the control input and its variability between consecutive time steps, respectively.

A critical issue of this control technique concerns the choice of parameters. Many of these can be selected based on the physics of the system to be controlled or appropriate control hardware limits, such as the actuation constraints or the control time step. The selection of parameters involved in (3), as well as the length of the prediction/control window, is traditionally tuned by trial and error. In § 2.1.4 we introduce our hyperparameter self-tuning procedure.

2.1.2. *Nonlinear system identification*

The optimization loop of MPC requires an accurate plant model for predicting the system's dynamics. The choice of the predictive model typically involves a trade-off between accuracy and computational complexity. In this work we adopt a data-driven sparsity-promoting technique. The framework was illustrated in Brunton *et al.* (2016*a*) and later applied in MPC by Kaiser *et al.* (2018) for a variety of nonlinear dynamical systems. It must be remarked that the self-tuning framework proposed here can easily be adapted to accommodate a different plant model, either analytical or data driven, including deep-learning models.

SINDy is a method that identifies a system of ordinary differential equations describing the dynamical system. In particular, the version described in this work corresponds to the extension of the SINDy model with exogenous input (SINDYc, Brunton *et al.*, 2016*b*). Considering a system of ordinary differential equations such as the one shown in (1), SINDYc derives an analytical expression of f from data. This process requires a dataset comprising the time series of the state vector $\boldsymbol{a}$ and the exogenous input $\boldsymbol{b}$. This method is based on the idea that most physical systems can be characterised by only a few relevant terms, resulting in governing equations that are sparse in a high-dimensional nonlinear function space. The resulting sparse model identification aims to find a balance between model complexity and accuracy, preventing overfitting of the model to the data.

To derive an expression of the function f from data, a discrete sampling is performed on a time vector, yielding r snapshots at time instances $t_i, i = 1, \ldots, r$ for the state vector $\boldsymbol{a}_i = \boldsymbol{a}(t_i)$, its time derivative $\dot{\boldsymbol{a}}_i = \dot{\boldsymbol{a}}(t_i)$ and the input signal $\boldsymbol{b}_i = \boldsymbol{b}(t_i)$. The data obtained are arranged into three matrices $\mathbf{A} \in \mathbb{R}^{r\times N_a}$, $\dot{\mathbf{A}} \in \mathbb{R}^{r\times N_a}$ and $\mathbf{B} \in \mathbb{R}^{r\times N_b}$:

$$\begin{aligned} \mathbf{A} &= (\boldsymbol{a}_1\,, \ldots\,, \boldsymbol{a}_r)'\,, \\ \dot{\mathbf{A}} &= (\dot{\boldsymbol{a}}_1\,, \ldots\,, \dot{\boldsymbol{a}}_r)'\,, \\ \mathbf{B} &= (\boldsymbol{b}_1\,, \ldots\,, \boldsymbol{b}_r)'\,. \end{aligned} \tag{5}$$

A library of functions $\boldsymbol{\Theta}$ is then set as

$$\begin{aligned} \boldsymbol{\Theta}(\mathbf{A},\mathbf{B}) = \; & (\mathbf{1}_r,\, \mathbf{A},\, \mathbf{B},\, ((\mathbf{A}_{1\bullet}\otimes\mathbf{A}_{1\bullet})', \ldots, (\mathbf{A}_{r\bullet}\otimes\mathbf{A}_{r\bullet})')'\,, \\ & ((\mathbf{A}_{1\bullet}\otimes\mathbf{B}_{1\bullet})', \ldots, (\mathbf{A}_{r\bullet}\otimes\mathbf{B}_{r\bullet})')' \\ & ((\mathbf{B}_{1\bullet}\otimes\mathbf{B}_{1\bullet})', \ldots, (\mathbf{B}_{r\bullet}\otimes\mathbf{B}_{r\bullet})')', \ldots), \end{aligned} \tag{6}$$

where $\mathbf{1}_r$ is a column vector of r ones, $\mathbf{A}_{i\bullet}$ denotes the i^{th} row of the matrix $\mathbf{A}$ and $\mathbf{H}\otimes\mathbf{K}$ denotes the Kronecker products of $\mathbf{H}$ and $\mathbf{K}$. The choice of functions to be included in the library is typically made by the user. This procedure is often guided by experience and prior knowledge about the dynamics of the system to be controlled. This issue introduces some limitations that are later discussed in §4.

The system data can be then assumed to be generated from the following model:

$$\dot{\mathbf{A}} = \mathbf{\Theta}(\mathbf{A}, \mathbf{B})\mathbf{\Xi}. \tag{7}$$

Here $\mathbf{\Xi} = \left(\boldsymbol{\xi}^1, \ldots, \boldsymbol{\xi}^{N_a}\right)$ represents the matrix whose rows are vectors of coefficients determining which terms of the right-hand side are active in the dynamics of the k^{th} state vector component. This matrix is sparse for many of the dynamical systems considered. It can be obtained by solving the optimization problem

$$\hat{\boldsymbol{\xi}}^k = arg \min_{\boldsymbol{\xi}^k} \left\{ \frac{1}{2} \left\| \dot{\mathbf{A}}_{\bullet k} - \mathbf{\Theta}(\mathbf{A}, \mathbf{B})\boldsymbol{\xi}^k \right\|_2^2 + \lambda \left\| \boldsymbol{\xi}^k \right\|_1 \right\}, \tag{8}$$

where λ is called the sparsity-promoting coefficient, $\mathbf{A}_{\bullet k}$ denotes the k^{th} column of the matrix $\mathbf{A}$. $\|\cdot\|_1$ and $\|\cdot\|_2$ are the L_1 and L_2 norms, respectively. It must be remarked that λ must be optimised to reach a compromise between parsimony and accuracy. Nonetheless, since this block of the MPC framework can easily be replaced by other strategies, λ will not be included in the self-tuning optimization illustrated in § 2.1.4.

Finally, the system can be described by

$$\dot{a}^k = \mathbf{\Theta}(\boldsymbol{a}, \boldsymbol{b})\hat{\boldsymbol{\xi}}^k, \quad k = 1, \ldots, N_a, \tag{9}$$

where in this case $\mathbf{\Theta}(\boldsymbol{a}, \boldsymbol{b})$ is a vector that takes into account the same functions included in the library in (6). The SINDYc-based model is utilised to make predictions of the dynamics involved in the control process starting from a specific state's initial condition. The steps described in this section to obtain the plant model are also condensed in the nonlinear system identification step in Algorithm 1.

2.1.3. *Local Polynomial Regression*

Effectively estimating system dynamics is crucial for various control techniques, particularly in MPC, where sensor feedback is utilised to make informed decisions. The accuracy of system dynamics estimation becomes paramount in MPC as it relies on sensor information to predict the behaviour of the controlled system. However, the presence of noise can hinder accurate estimation, necessitating the implementation of noise mitigation techniques.

This challenge falls within the broader context of time series analysis, as sensor outputs represent discrete samples over time of a process variable. To address measurement noise, LPR emerges as a robust, accurate and cost-effective non-parametric smoothing technique. Applying LPR to sensor output data enhances the reliability and accuracy of estimating the current state of the system under varying noise conditions. In this section we provide a succinct overview of the formulation of LPR. The notation convention involves denoting random variables with capital letters, while deterministic values are represented in lowercase.

The objective is to predict or explain the response S (sensor output) using the predictor T (time) from a sample $\{(T_j, S_j)\}_{j=1}^n$ using the following regression model:

$$S_j = m(T_j) + \sigma(T_j)\epsilon_j, \tag{10}$$

Here m is the regression function, $\{\epsilon_j\}_{j=1}^n$ are zero mean and unit variance random variables and σ^2 is the point variance. For simplicity, it is assumed that the model is homoscedastic, i.e. that the variance is constant. From a practical point of view, σ also represents the measurement of noise intensity.

The reason why LPR is used in this context is that it allows for a fast joint estimation of the regression function and its derivatives without making any prior. By making only a few regularity assumptions, it is characterised by a good flexibility that allows us to model complex relations beyond a parametric form. The working principle of the LPR is to approximate the regression function locally by a polynomial of order p, performing a weighted least squares regression using data only around the point of interest.

It is assumed that the regression functions admit derivatives up to the order $p+1$ and, thus, that it can be expanded in a Taylor's series. For close time instants t and T_j, the unknown regression function can be approximated by a polynomial of order p, then

$$m(T_j) \approx \sum_{i=0}^{p} \frac{m^{(i)}(t)}{i!}(T_j - t)^i \equiv \sum_{i=0}^{p} \beta_i (T_j - t)^i, \tag{11}$$

The vector of parameters $\boldsymbol{\beta} = (\beta_0, \ldots, \beta_p)'$ can be estimated by solving the minimization problem

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \sum_{j=1}^{n} \{S_j - \sum_{i=0}^{p} \beta_i (T_j - t)^i\}^2 K_h(T_j - t), \tag{12}$$

where h is the bandwidth determining the size of the local neighbourhood (also called smoothing parameter) and $K_h(t) = \frac{1}{h}K(\frac{t}{h})$ with K a kernel function assigning the weights to each observation. Once the vector $\hat{\boldsymbol{\beta}}$ has been obtained, an estimator of the q^{th} derivative of the regression function is

$$\hat{m}^{(q)}(t) = q!\hat{\beta}_q. \tag{13}$$

The solution of the optimization in (12) is simplified when the methodology is presented in matrix form. To that end, $\mathbf{T} \in \mathbb{R}^{n\times(p+1)}$ is the design matrix

$$\mathbf{T} = \begin{pmatrix} 1 & (T_1 - t) & \ldots & (T_1 - t)^p \\ \vdots & \vdots & & \vdots \\ 1 & (T_n - t) & \ldots & (T_n - t)^p \end{pmatrix} \tag{14}$$

while the vector of responses is $\boldsymbol{S} = (S_1, \ldots, S_n)'$ and $\mathbf{W} = \mathrm{diag}(K_h(T_1 - t), \ldots, K_h(T_n - t))$. According to this notation and assuming the invertibility of $\mathbf{T}'\mathbf{W}\mathbf{T}$, the weighted least squares solution of the minimization problem expressed in (12) is

$$\hat{\boldsymbol{\beta}} = (\mathbf{T}'\mathbf{W}\mathbf{T})^{-1}\mathbf{T}'\mathbf{W}\boldsymbol{S} \tag{15}$$

and, thus, the estimator of the local polynomial is

$$\hat{m}(t) = \boldsymbol{e}_1'(\mathbf{T}'\mathbf{W}\mathbf{T})^{-1}\mathbf{T}'\mathbf{W}\boldsymbol{S} = \sum_{j=1}^{n} W_j^p(t) S_j, \tag{16}$$

where $\boldsymbol{e}_k \in \mathbb{R}^{p+1}$ a vector having 1 in the k^{th} entry and zero elsewhere and $W_j^p(t) = \boldsymbol{e}_1'(\mathbf{T}'\mathbf{W}\mathbf{T})^{-1}\mathbf{T}'\mathbf{W}\boldsymbol{e}_j$.

The critical element that determines the degree of smoothing in LPR is the size of the local neighbourhood. The selection of h determines the accuracy of the estimation of the regression function: too large values lead to a large bias in the regression, while too small values increase the variance. Its choice is generically the result of a trade-off between the variance and the estimation bias of the regression function. There are several methods to select the bandwidth to be used for LPR. One possible approach is to choose between a global bandwidth, which is common to the entire domain and is optimal for the entire range of data, or a local bandwidth that depends on the covariate and is therefore optimal at each point of the regression function estimation. The latter would allow for more flexibility in estimating inhomogeneous regression functions. However, in the proposed framework, a global bandwidth was chosen due to its simplicity and, more importantly, its reduced computational cost. In the presented approach, the global bandwidth is chosen using the leave-one-out cross-validation methodology. Specifically, this parameter corresponds to the one that minimises the following expression:

$$\sum_{j=1}^{n} \left(S_j - \hat{m}_{h,-j}(T_j)\right)^2 . \tag{17}$$

Here $\hat{m}_{h,-j}(T_j)$ denotes the estimation of the regression function while excluding the j^{th} term. For further information regarding the optimal bandwidth selection, see Wand & Jones (1994) and Fan & Gijbels (1996). Additionally, there exist other methodologies that utilise machine-learning techniques to select the local bandwidth such as in Giordano & Parrella (2008) where neural networks are used.

Once the smoothing parameter has been set, it remains to choose the weighting function and the degree of the local polynomial, although these two have a minor influence on the performance of the LPR estimation. A common choice for the first is the Epanechnikov kernel. As for the degree of the local polynomial, there is a general pattern of increasing variability according to which, to estimate $m^{(q)}(t)$, the lowest odd order is recommended, i.e. $p = q + 1$ or occasionally $p = q + 3$ (Fan & Gijbels, 1996).

It is also worth noting that due to asymmetric estimation within the control algorithm, boundary effects arise, leading to a bias in the estimation at the edge. These effects, discussed in more detail in Fan & Gijbels (1996), disappear when using local linear regression ($p = 1$).

2.1.4. *Hyperparameter automatic tuning with BO*

Model predictive control relies on a specific set of hyperparameters to precisely define the cost function in (2) for the selection of the optimal action over the control window. A new functional, based on the global performance of the control, is defined. Bayesian optimization is employed to find the hyperparameter vector that maximises the control performance. By adopting this approach, a more efficient and effective MPC framework may be achieved. The hyperparameters are indeed adapted to different conditions of measurement noise and/or model uncertainty by the BO process. This proposed framework is practically independent of the user.

The parameters under consideration include *i)* the components of the weight matrices, as presented in (3), which penalise errors of the state vector with respect to the control target trajectories, input cost and input time variability; and *ii)* the length of the control/prediction windows, assumed equal for simplicity. All aforementioned control parameters are included in a single vector denoted as $\boldsymbol{\eta} \in \mathbb{R}^{N_{\boldsymbol{\eta}}}$, where $N_{\boldsymbol{\eta}} = 2N_b + N_c + 1$. A further reduction in the number of control parameters can also be considered by imposing that the components of $\mathbf{R}_b$ and $\mathbf{R}_{\Delta b}$ related to the rear cylinders of the fluidic pinball are identical under a flow symmetry argument, as in Bieker *et al.* (2020). Therefore, in this case it is stated that $R_{b^2} = R_{b^3} = R_{b^{2,3}}$ and $R_{\Delta b^2} = R_{\Delta b^3} = R_{\Delta b^{2,3}}$ and the total number of parameters is reduced to $N_{\boldsymbol{\eta}} = 2(N_b - 1) + N_c + 1$. Section 3 also provides several results that justify this choice.

Note that control results depend on the selection of the vector $\boldsymbol{\eta}$. Consequently, an offline optimization process can be implemented to select the optimal value of $\boldsymbol{\eta}$, which maximises the control performance. Consider using a specific realization of the hyperparameter vector $\boldsymbol{\eta}$ to control the system. The state vector measure is indirectly dependent on the choice of hyperparameter vector and is denoted here by $\tilde{\boldsymbol{s}}(\boldsymbol{\eta})$. By running the control on a discrete-time vector of n_{BO} time steps, the cost function can be defined as

$$\mathcal{J}_{BO}(\boldsymbol{\eta}) = \frac{1}{n_{BO}} \sum_{k=1}^{N_c} \sum_{j=1}^{n_{BO}} \left(\tilde{s}_j^k(\boldsymbol{\eta}) - c_*^k\right)^2 . \tag{18}$$

Note that the user selects the target to be optimised, specified by the cost function $\mathcal{J}_{BO}$. In addition, in absence of measurement noise, $\tilde{\boldsymbol{s}}(\boldsymbol{\eta})$ is the ideal measure of the target feature vector. Otherwise, the LPR technique is applied to the state vector measure. In this study the parameters are optimised to minimise the quadratic error between the controlled variables and the user-defined target. Thus, the optimal hyperparameter vector can be obtained by solving the problem

$$\boldsymbol{\eta}_{opt} = \underset{\boldsymbol{\eta} \in H}{\arg\min} \quad \mathcal{J}_{BO}(\boldsymbol{\eta}), \tag{19}$$

where $H \subset \mathbb{R}^{N_{\boldsymbol{\eta}}}$ is the search domain. Solving the optimization in (19) is computationally costly due to the expensive sampling of the black-box cost function $\mathcal{J}_{BO}$. Each sample of $\mathcal{J}_{BO}$ is obtained via application of the MPC over n_{BO} time steps. In this framework, BO serves as an efficient method to address this problem, offering an algorithm to search the minimum of the function with a high guarantee of avoiding local minima and characterised by rapid convergence. Indeed, BO has shown to be an efficient strategy particularly when $N_{\boldsymbol{\eta}} \leq 20$ and the search domain H is a hyper-rectangle, that is $H = \{\boldsymbol{\eta} \in \mathbb{R}^{N_{\boldsymbol{\eta}}} | \eta^i \in [\eta^i_{min}, \eta^i_{max}] \subset \mathbb{R}\,, \eta^i_{min} < \eta^i_{max}\,, i = 1, \ldots, N_{\boldsymbol{\eta}}\}$, where $\boldsymbol{\eta}_{min}$ and $\boldsymbol{\eta}_{max}$ are the lower and upper bound vectors of the hyper-rectangle, respectively.

In order to find the minimum of the unknown objective function, BO employs an iterative approach, as shown in the MPC-tuning step of Algorithm 1. It utilises a probabilistic model, typically a Gaussian process (GP), to estimate the behaviour of the objective function. At each iteration, the probabilistic model incorporates available data points to make predictions about the function behaviour at unexplored points in the search space. Simultaneously, an acquisition process guides the samplings by suggesting the optimal locations in order to discover the minimum point. A specific function (called the acquisition function) is set to balance exploration, by directing attention to less-explored areas of the search domain, and exploitation, by concentrating on regions near potential minimum points. Finally, the BO iterations continue until a stopping criterion is met, such as reaching a maximum number of iterations or achieving convergence in the search for the minimum. Thus denoting as $\alpha(\boldsymbol{\eta})$ the acquisition function selected, the point where to sample next in the iterative approach, that is, $\boldsymbol{\eta}^+$, can be obtained by solving

$$\boldsymbol{\eta}^+ = \underset{\boldsymbol{\eta} \in H}{\arg\max}\ \alpha(\boldsymbol{\eta}). \tag{20}$$

The expected improvement is used as an acquisition function (Snoek *et al.*, 2012), which evaluates the potential improvement over the current best solution.

The subsequent discussion will centre on the probabilistic model utilised in BO. Specifically, a prior distribution, which corresponds to a multivariate Gaussian distribution, is employed. Consider the situation where problem (19) is tackled using BO. A GP regression is required for the functional $\mathcal{J}_{BO}$ based on the available observations of the function up to the given iteration.

Since $\mathcal{J}_{BO}$ is a Gaussian process, for any collection of D points, included in $\boldsymbol{\Xi} = (\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_D)'$, then the vector of function samplings at these points, denoted as $\boldsymbol{J} = (\mathcal{J}_{BO}(\boldsymbol{\eta}_1), \ldots, \mathcal{J}_{BO}(\boldsymbol{\eta}_D))'$ is multivariate-Gaussian distributed, then

$$\boldsymbol{J} \sim \mathcal{N}_D(\boldsymbol{\mu}, \mathbf{K}) \tag{21}$$

where $\boldsymbol{\mu} = (\mu(\boldsymbol{\eta}_1), \ldots, \mu(\boldsymbol{\eta}_D))'$ is the mean vector and $\mathbf{K}$ the covariance matrix whose ij^{th} component is $K_{ij} = k(\boldsymbol{\eta}_i, \boldsymbol{\eta}_j)$, with μ a mean function and k a positive definite kernel function. For simplicity, the mean function is assumed to be null.

In order to make a prediction of the value of the unknown function at a new point of interest $\boldsymbol{\eta}_*$, denoted as $\mathcal{J}_{BO_*}$, conditioned on the values of the function already observed, and included in the vector $\boldsymbol{J}$, the joint multivariate Gaussian distribution can be considered:

$$\begin{pmatrix} \boldsymbol{J} \\ \mathcal{J}_{BO_*} \end{pmatrix} \sim \mathcal{N}_{D+1}\left(\mathbf{0}, \begin{pmatrix} \mathbf{K} & \boldsymbol{k}_* \\ \boldsymbol{k}_*' & k_{**} \end{pmatrix}\right). \tag{22}$$

Here $k_{**} = k(\boldsymbol{\eta}^*, \boldsymbol{\eta}^*)$ and the vector $\boldsymbol{k}_* = (k(\boldsymbol{\eta}_1, \boldsymbol{\eta}_*), \ldots, k(\boldsymbol{\eta}_D, \boldsymbol{\eta}_*))'$. This equation describes how the samples $\boldsymbol{J}$ at the locations $\boldsymbol{\Xi}$ correlate with the sample of interest $\mathcal{J}_{BO_*}$, whose conditional distribution is

$$\mathcal{J}_{BO_*} | \boldsymbol{\eta}_*, \boldsymbol{\Xi}, \boldsymbol{J} \sim \mathcal{N}\left(\mu_*\,,\, \sigma_*^2\right) \tag{23}$$

with mean $\mu_* = \boldsymbol{k}_*' \boldsymbol{K}^{-1} \boldsymbol{J}$ and variance $\sigma_*^2 = \left(k_{**} - \boldsymbol{k}_*' \boldsymbol{K}^{-1} \boldsymbol{k}_*\right)^2$. Equation (23) provides the posterior distribution of the unknown function at the new point where the sample has to be performed. It then furnishes the surrogate model used to describe the function $\mathcal{J}_{BO}$ in the domain for the search of the minimum point.

2.2. *The fluidic pinball*

The proposed framework is tested on the control of the two-dimensional viscous incompressible flow around a three-cylinder configuration, commonly referred to as fluidic pinball (Pastur *et al.*, 2019; Deng *et al.*, 2020, 2022). The fluidic pinball was chosen because it represents a suitable multiple-input-multiple-output system benchmark for flow controllers.

A representation of the fluidic pinball can be seen in Figure 3. The three cylinders have identical diameters $D = 2R$, and their geometric centers are placed at the vertices of an equilateral triangle of side $3R$. The centres are symmetrically positioned with respect to the direction of the main flow. The leftmost vertex of the triangle points upstream while the rightmost side is orthogonal to the flow direction. The free stream has a constant velocity equal to U_∞. The cylinders of the fluidic pinball, denoted here with 1 (front), 2 (top) and 3 (bottom), can rotate independently around their axes (orthogonal to the plane of the flow) with tangential velocity b^1, b^2 and b^3, respectively.

The dynamics of the wake past the fluidic pinball is obtained through a two-dimensional direct numerical simulation (DNS) with the code developed by Noack & Morzyński (2017).

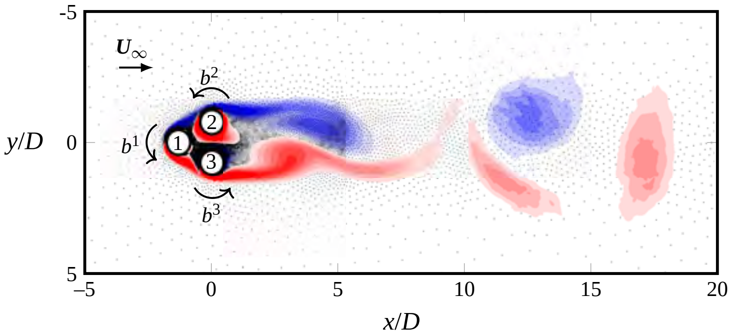

Figure 3: Domain of the incompressible two-dimensional DNS of the flow around the fluidic pinball. Front, top and bottom cylinders are labelled as 1, 2 and 3, respectively. The rotational velocities of the cylinders are b^1, b^2 and b^3. The arrows indicate positive (counterclockwise) rotations. The background shows the 8633 nodes grid used for the DNS. The contour colours indicate the out-of-plane vorticity.

The flow is described in a Cartesian reference system in which the x and y axes are in the streamwise and crosswise directions, respectively. The centre of the Cartesian reference system coincides with the mid-point of the rightmost bottom and top cylinder and the computational domain, that is, Ω, is bounded in $(-5D, 20D) \times (-5D, 5D)$. The position in the reference system is then described by the vector $\boldsymbol{x} = (x, y) = x\boldsymbol{e_1} + y\boldsymbol{e_2}$, where $\boldsymbol{e_1}$ and $\boldsymbol{e_2}$ are respectively the unit vectors in the directions of the x and y axes and the velocity vector is assumed to be $\boldsymbol{u} = (u, v)$. The constant density is denoted by ρ, the kinematic viscosity of the fluid by ν. All quantities used in the discussion are assumed non-dimensionalised with cylinder diameter, free-stream velocity, and fluid density. The Reynolds number is defined as $Re_D = \frac{U_\infty D}{\nu}$. A value of $Re_D = 150$ is adopted, which is sufficiently large to ensure a chaotic behaviour, although still laminar. The two-dimensional solver has already been used in previous work at this same Re_D (Wang *et al.*, 2023). The boundary conditions comprise a far-field condition ($\boldsymbol{u} = U_\infty \boldsymbol{e_1}$) in the upper and lower edges, a stress-free one in the outflow edge, and a no-slip condition on the cylinder walls, which in the absence of forcing becomes $\boldsymbol{u} = 0$.

The DNS allows forcing by independent rotation of the cylinders. To this purpose, a velocity with module $|b^i|$ is imposed at the cylinder surface. Positive values of b^i are associated with counterclockwise rotations of the cylinders. In the remainder of the paper, a reference time scale is set as the convective unit (*c.u.*), i.e. the time scale based on the freestream velocity and the cylinder diameter. The lift coefficient is defined as $C_l = \frac{2F_l}{\rho U_\infty^2 D}$, where F_l is the total lift force applied to the cylinders in the direction of the y axis and the same quantity is applied in the scaling of F_d, force applied to the cylinders in the direction of the x axis, to obtain the total drag coefficient C_d. The DNS uses a grid of 8633 vertices and 4225 triangles accounting for both accuracy and computational speed. A preliminary grid convergence study at $Re_D = 150$ identified this as sufficient resolution for errors of up to 3% in the free case and $\approx 4\%$ in the actuated case in terms of drag and lift.

2.3. *The control approach*

The aim of the control is to achieve a reduction in the overall drag coefficient of the fluidic pinball, while also controlling the lift coefficient so that the latter has reduced oscillations with a zero average value. The vector of target features is therefore composed only of the total lift and drag coefficients, $\boldsymbol{c} = (C_d, C_l)'$. The target vector is thus composed of null components $\boldsymbol{c}_* = (0, 0)'$. Since the cost function in (2) is quadratic, a null target vector will penalise both mean values of C_l and C_d and their oscillations.

To this purpose, the tangential velocities of the three cylinders are tuned respecting the implementation limits, here chosen equal to $b^i \in [-1, 1]$ and $\Delta b^i \in [-4, 4]$, for all$i = 1, 2, 3$. The model plant has a state vector composed by C_d and C_l and their time derivatives, respectively $\dot{C}_d$ and $\dot{C}_l$ (i.e. $\boldsymbol{a} = (C_d, C_l, \dot{C}_d, \dot{C}_l)'$). A state vector based solely on global variables does not require adding intrusive probes for flow estimation. Similar state vector choices that incorporate aerodynamic force coefficients and temporal derivatives are made in Nair *et al.* (2019) and Loiseau *et al.* (2018). While this approach is often effective in separated flows, it might be challenged at higher *Re* flows and of unfeasible application in other flow configurations.

Feedback to the control involves measurement of drag and lift forces, so at each time instant t_j, it is considered $\boldsymbol{s}_j = \boldsymbol{c}_j$. This approach also facilitates a reduction in the complexity of the predictive model, thereby speeding up the control process. Indeed, a compact state vector is particularly desirable to reduce the computational cost of the iterative optimization problem over receding horizons of MPC.

In the present work, noise in lift and drag measurement is also considered. Under this assumption, the model in (10) is applied. The noise intensity σ is assumed to be constant over time and given as a percentage of the full-scale measured drag and lift coefficients in conditions without actuation. Therefore, in the case of measurement noise in the sensors, the response variable to which the LPR is applied corresponds to the measurements of the drag and lift coefficients of the fluidic pinball. The LPR enables us to obtain estimates of their regression function in output from the sensors but also of their time derivatives, allowing us to use this information as control feedback.

The MPC-optimization problem is solved every control time step (so every T_s) to update the exogenous control input. During the time between two consecutive samples, the control input to the system remains constant. Thus, the sampling time step should be chosen small enough to ensure a good closed-loop performance, but not too small to avoid an excessive computational cost. In this work, it is set at $T_s = 0.5\,c.u.$, i.e. sufficiently small compared with the shedding period of the fluidic pinball wake (denoted as T_{sh}) and not too high to affect control performance. The reason why T_s was not included in the control tuning concerns the difficulty of defining the search domain for realistic applications. Imposing bounds on this parameter requires an estimate of the computational cost of the MPC, and this procedure is postponed to future experimental applications. The method used to optimise the control action in the MPC framework is sequential quadratic programming with constraints. The optimization was carried out with a built-in function of MATLAB. The stop criteria are set at a maximum number of iterations of 500 and a step tolerance of $1e{-6}$.

At each measurement taken during the control process, in the presence of noise in the C_l and C_d sensors, the LPR technique is applied to a time series of outputs of length corresponding to the characteristic shedding period of the fluidic pinball wake, as observed in § 3. Local polynomial regression acts asymmetrically, i.e. only past output sensor data are available.

Parameter tuning is performed by evaluating the cost function $\mathcal{J}_{BO}$ over a time history of the controlled variables of one shedding period of the fluidic pinball. The transient phase is excluded. Finally, as a stopping criterion for the search for the optimal hyperparameter vector, a maximum number of 100 iterations in BO was set. This proved sufficient to reach convergence in the tuning process, as shown in § 3.2.

3. Results

In this section the results of the current work are presented. Subsection 3.1 outlines the prediction performance of the SINDYc-based force model. Furthermore, the prediction performance enhancement when using LPR in the presence of measurement noise is illustrated. In § 3.2 the effectiveness of hyperparameters tuning in the different control scenarios is examined. Lastly, in § 3.3 the outcomes of applying the MPC algorithm to the fluidic pinball wake for drag reduction purposes under ideal measurement conditions and in the presence of sensor noise are presented. An additional discussion is provided to justify the choice of symmetry in the parameters of the rear cylinders of the fluidic pinball.

3.1. *Predictive model*

This subsection presents the performance evaluation of the predictive model employed in the control framework. SINDYc is applied on a training dataset consisting of a time series of the system state, along with predetermined actuation laws. The chosen open-loop laws are composed of step signals for the three-cylinder velocities b^i, encompassing all possible combinations of the velocities of the three cylinders within control constraints. Specifically, velocities ranging from -1 to 1 with an increment of 0.5 are explored, resulting in a total of 5^3 combinations. For each explored velocity combination, the actuation steps are long enough ($\approx 55\,c.u.$) to allow reaching the respective steady state. The total length of the generated dataset is $6950\,c.u.$. The actuation time series is subsequently smoothed to include the transient dynamics in the force model. It must be noted that this approach is effective for this test case whose dynamics evolve on a clearly defined low-dimensional attractor. Open-loop training for plant identification has been shown effective in separated flows (see, e.g. Kaiser *et al.*, 2017; Bieker *et al.*, 2020). More complex nonlinear dynamical systems might require different strategies for adequate training of the modelling plant.

Additionally, a library of polynomial functions up to the second order is chosen for the sparse regression with SINDYc.

Figure 4 shows the prediction performance of the plant model. The presented results were generated by performing predictions on a testing dataset with a preassigned smooth law of the rotational speeds of the three cylinders. Predictions of $4\,c.u.$ using initial conditions at random points of the validation dataset, and for a statistically significant number of times were performed. The resulting time series of each prediction were then used to calculate their respective errors ($\hat{e}_{C_d}$ for C_d and $\hat{e}_{C_l}$ for C_l). The plot illustrates the trend of the average and the confidence region for σ of the prediction errors, respectively normalised with the standard deviation of C_d and C_l in the free response, here denoted as $sd(C_{d_0})$ and $sd(C_{l_0})$, respectively.

The panels on the left show the prediction performance calculated from an initial condition obtained in the absence of measurement noise in the sensors. The average value

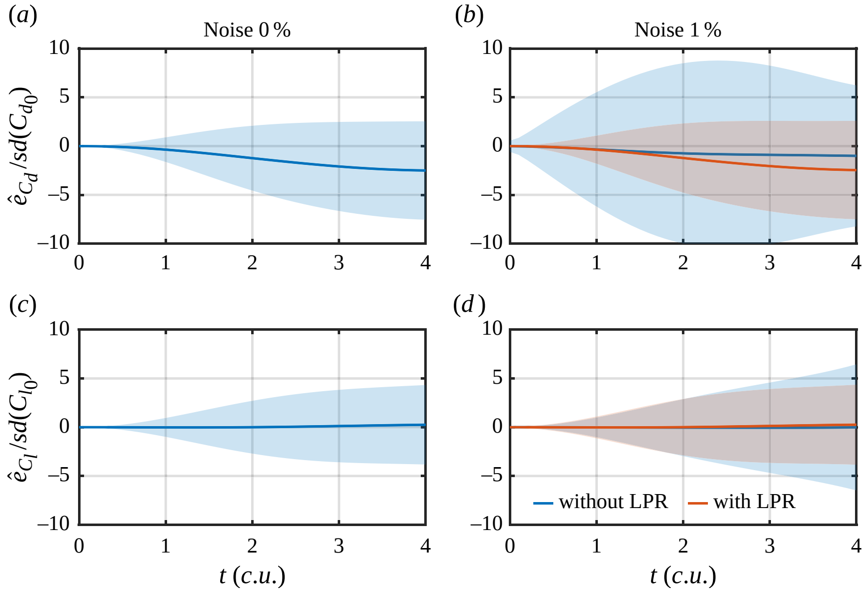

Figure 4: Prediction performance of the force model obtained using SINDYc. The plots display the average value and the confidence region for σ of the probability distribution of the normalized C_d and C_l prediction errors with respect to their standard deviation under unforced conditions. Panels (a, c) show the results in absence of measurement noise of the initial condition while (b, d) with 1% of measurement noise. In the presence of noise, a comparison is shown with the results of predictions obtained using LPR estimation as an initial condition.

of the normalised error distribution maintains values close to zero for any length of the prediction window analysed. Furthermore, for short prediction lengths (up to $\approx 3\,c.u.$) the confidence bounds for σ are still confined within a narrow region. In this context, as they are not directly measurable, the time derivatives of the aerodynamic coefficients are calculated using finite differences. This approach, however, is highly sensitive to measurement noise. The non-parametric smoothing technique LPR allows joint estimation of the output data from the sensors and their respective time derivatives. The panels on the right show the benefits brought by initial-condition setting with LPR to the prediction performance in the presence of a sensor measurement noise of 1%. These plots show a comparison of the statistical error distributions for two distinct cases: in one the initial condition corresponds to the LPR estimation of the data with noise, in the other, it is not used. It is of considerable interest to observe that the prediction with LPR performs similarly to the case without noise. The figure discussed above could be used to choose the MPC prediction window length as it provides an idea of the uncertainty propagation in the prediction horizon. However, since this parameter has a significant effect on control results in terms of performance and computational cost, it is left to automatic selection through BOn, so as to also take into account the effects of measurement noise in its selection.

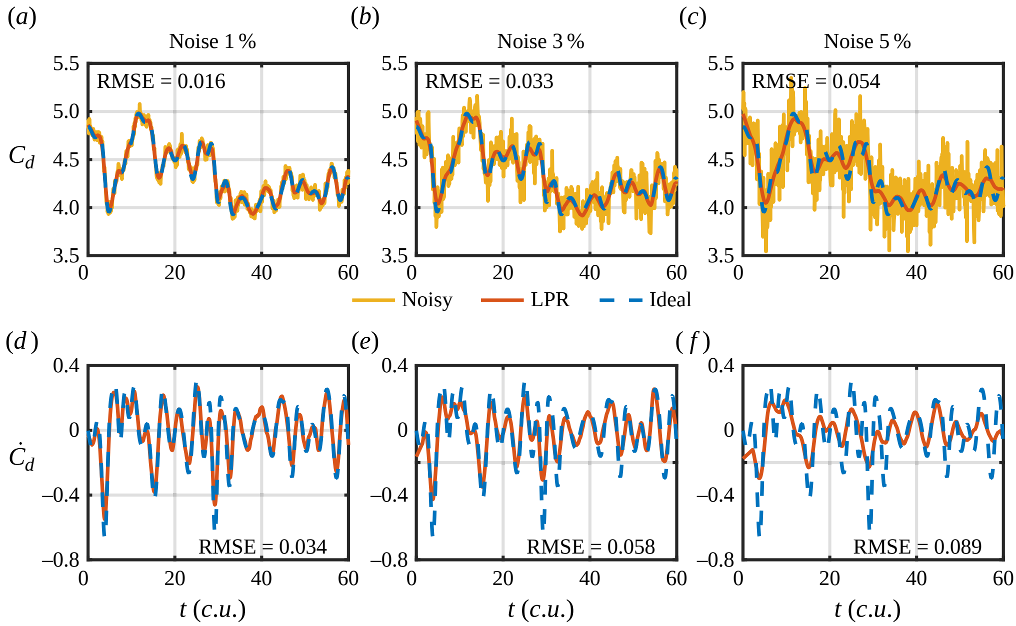

Figure 5: Local polynomial regression applied to a C_d time series within a time period of $60c.u.$. This case corresponds to forced fluidic pinball dynamics. Panels $(a-c)$ display C_d estimation in the presence of increasing noise levels $(1\%, 3\%, 5\%)$. Panels $(d-f)$ show the LPR estimation of $\dot{C}_d$. Each plot includes the ideal and noisy time series, also including the RMSE of the LPR estimation.

In addition, Figure 5 provides further insights into the advantages of LPR. In this case the estimation of C_d and $\dot{C}_d$ is tested on a case with external forcing and under increasing noise level. The trend in the LPR estimation error of both C_d and $\dot{C}_d$ is illustrated as the noise increases. The estimation errors are evaluated using a root-mean-square error (RMSE) of the estimate compared with the ideal data without noise.

The results demonstrate the effectiveness of LPR in estimating C_d and $\dot{C}_d$ with high accuracy even in the presence of high levels of noise. Thus, the benefits brought by LPR enable the control feasibility even in high noise scenarios.

3.2. *Bayesian optimization for tuning control parameters*

This subsection presents the results of the hyperparameters tuning using the BO algorithm. Various control scenarios are analysed, including clean, low and high noise level cases.

The set of hyperparameters to be optimised is composed of the elements of the weight matrices in (3) and the length of the control/prediction window, used in the definition of the cost functional in (2). Specifically, the components of the matrix $\mathbf{Q}$ are optimised within the interval $[0.1, 5]$, the terms of $\mathbf{R}_b$ and $\mathbf{R}_{\Delta b}$ in $[0.1, 10]$ and w_p is optimised within $[1, 4]$.

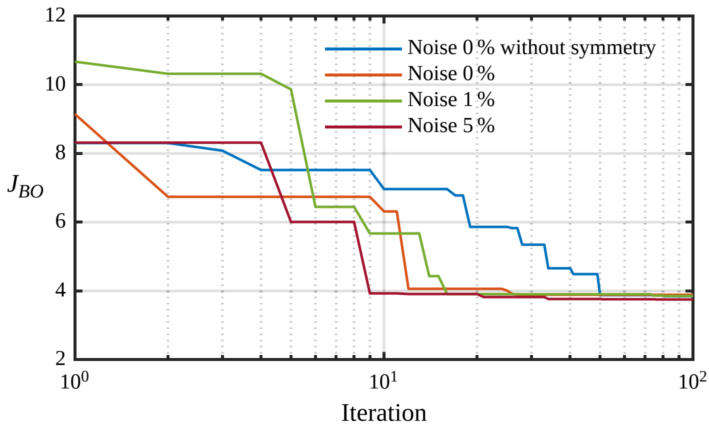

Figure 6: Minimum observed J_{BO} sampling history during MPC hyperparameter tuning. The x axis of the plot is in logarithmic scale. The results of some control scenarios in the presence and absence of measurement noise (and symmetry in the parameters) are presented.

Figure 6 displays the trend of the minimum observed $\mathcal{J}_{BO}$ samplings as the tuning process iterations progress for some of the analysed control cases. The results indicate that the optimization function ($\mathcal{J}_{BO}$) reaches convergence in less than 30 iterations for almost all cases. The tuned control parameters of each case are presented in Table 1. The table also provides the average values of C_d and C_l ($\bar{C}_d$ and $\bar{C}_l$), as well as their corresponding standard deviations ($sd(C_d)$ and $sd(C_l)$), during the control phase. These values are computed over a shedding time interval.

It must be remarked that the final set of hyperparameters for each case is run-dependent. This is mostly to be ascribed to the weak dependence of $\mathcal{J}_{BO}$ to some of the imposed constraints in the MPC loss function. Nonetheless, the differences in terms of drag reduction have been observed to be smaller than $\pm 1.1\%$ in different runs.

Figure 7 presents a plot of the cost function $\mathcal{J}_{BO}$ and its contributions due to the control of C_d (denoted as $\mathcal{J}^1_{BO}$) and C_l (denoted as $\mathcal{J}^2_{BO}$), varying Q^1 and Q^2, and then R_{b^1} and $R_{b^2} = R_{b^3} = R_{b^{2,3}}$ under 1% measurement noise. The two contributions of $\mathcal{J}_{BO}$ can be obtained by considering the cost functional in (18) and selecting only the component for $k = 1$ (to obtain $\mathcal{J}^1_{BO}$) and, for $k = 2$, to obtain $\mathcal{J}^2_{BO}$. The presented cost function plot is useful for interpreting the convergence parameters of the optimization process according to BO. The term $\mathcal{J}^2_{BO}$ has a minor influence on the optimization process, being approximately two orders of magnitude smaller than $\mathcal{J}^1_{BO}$. Therefore, the optimization process is mainly driven by the influence of the control parameters on reducing the C_d of the fluidic pinball.

As Q^1 increases, both $\mathcal{J}_{BO}$ and $\mathcal{J}^1_{BO}$ functions decrease sharply at first and then reach a plateau. Conversely, $\mathcal{J}^2_{BO}$ shows the opposite behaviour. A good C_l control would be achieved with a low Q^1 and high Q^2, but this would lead to a poor drag reduction. On the other hand, the choice of Q^2 has little impact on control performance, as the cost function $\mathcal{J}_{BO}$ remains nearly constant with its variation.

Noise	LPR	$diag(\mathbf{Q})$	diag($\mathbf{R}_b$)	diag($\mathbf{R}_{\Delta b}$)	w_p $(c.u.)$	$\bar{C_d}$	$sd(C_d)$	$\bar{C_l}$	$sd(C_l)$
0%	no	(1.000, 1.000)	(1.000, 1.000, 1.000)	(1.000, 1.000, 1.000)	3.0	3.0598	0.0956	-0.0530	0.1335
0%	no	(4.703, 0.206)	(6.334, 3.735, 1.147)*	(1.754, 0.176, 0.768)*	3.4	1.9685	0.0099	-0.2278	0.0687
0%	no	(4.881, 0.435)	(5.133, 0.903, 0.903)	(8.409, 0.193, 0.193)	3.2	1.9633	0.0094	0.0022	0.0604
0.5%	no	(4.886, 0.191)	(7.435, 0.514, 0.514)	(3.961, 0.465, 0.465)	3.8	1.9373	0.0200	0.0042	0.0863
0.5%	yes	(3.836, 2.283)	(9.776, 0.337, 0.337)	(0.367, 1.465, 1.465)	3.9	1.9294	0.0099	0.0172	0.0909
1%	yes	(4.938, 1.753)	(5.960, 0.170, 0.170)	(1.123, 2.521, 2.521)	4.0	1.9284	0.0091	0.0103	0.0914
5%	yes	(4.659, 1.007)	(9.556, 0.547, 0.547)	(0.135, 1.044, 1.044)	3.0	1.9371	0.0100	0.0110	0.0970

Table 1: Results of the control tuning through Bayesian optimization. A comparison is provided for all the analyzed simulation cases, with a particular focus on the tuning effects due to the intensity of the sensor measurement noise, the use of the LPR technique in the sensor outputs, and the imposition of symmetry in the control parameters. The asterisk (*) denotes control cases in which the tuning is performed on all control parameters, without imposing the symmetry condition on the parameters of the rear cylinders of the fluidic pinball. The table also provides the case with all parameters equal to 1 (except for the prediction/control window set at $3 c.u.$)

By observing Table 1 and considering the behaviour of the cost function, it is justified why almost all proposed control cases have Q^1 very close to the maximum achievable value, while there is greater variability in the selection of Q^2.

Similarly, due to the fact that the drag reduction drives the optimization process, the parameter that has the greatest influence on $\mathcal{J}_{BO}$ is $R_{b^{2,3}}$. A lower value of $R_{b^{2,3}}$ would assign less weight to the actuation cost of the rear cylinders of the fluidic pinball, allowing for greater rotation intensity. As will be explained in § 3.3.1, the main contributors to drag reduction are indeed the rear cylinders. On the other hand, the parameter R_{b^1} has little impact on the overall control performance since the front cylinder is responsible for the stabilization of the lift coefficient. These latter considerations justify that the optimum $R_{b^{2,3}}$ is achieved at low values, close to the lower possible limit, while greater variability is observed in the optimization of the R_{b^1} parameter.

3.3. *Application of the MPC algorithm*

This subsection shows the results of applying the control algorithm to the wake of the fluidic pinball. It is specified that all results reported from now on are obtained by imposing symmetry in the parameters of the rear cylinders.

3.3.1. *Control simulations without measurement noise*

Before discussing the effects of MPC, a brief description of the behaviour of the unforced wake of the fluidic pinball in the laminar chaotic regime, characteristic of the chosen Reynolds number ($Re_D = 150$), is presented.

Time histories of lift (plot a) and drag (plot b) coefficients for $500\, c.u.$ are reported in Figure 8. The figure also includes a power spectral density (PSD) of the lift coefficient (plot c), with the Strouhal number $St_D = \frac{fD}{U_\infty}$ related to the diameter of the three cylinders,

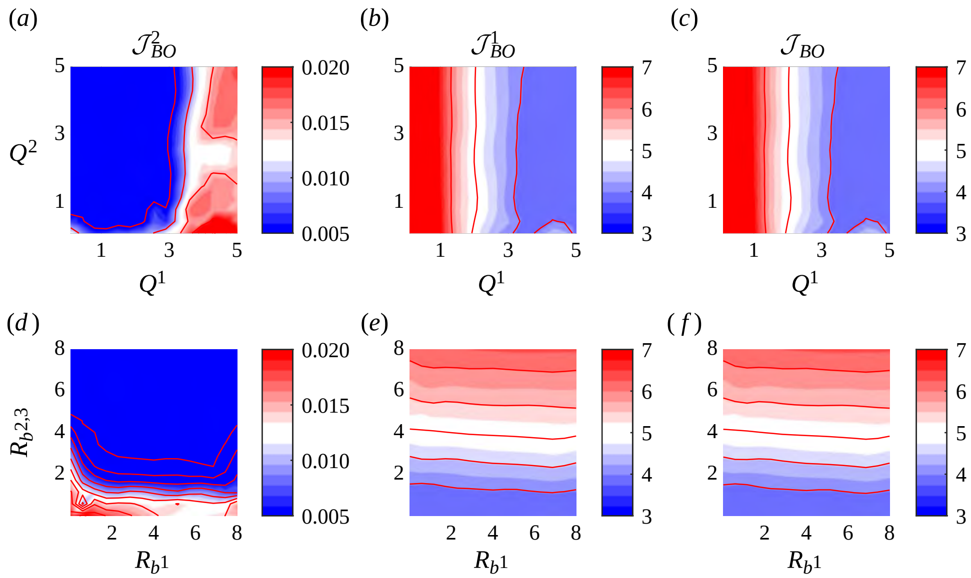

Figure 7: Representation of the cost functional J_{BO} in equation (18), the component due to C_l control ($\mathcal{J}^2_{BO}$) and the component due to C_d control ($\mathcal{J}^1_{BO}$). Contour plots are presented as functions of Q^1 and Q^2 $(a-c)$ and R_{b^1} and $R_{b^2} = R_{b^3} = R_{b^{2,3}}$ $(d-f)$. The control case considers a 1% measurement noise and symmetric weighting coefficients for the rear cylinders of the fluidic pinball. In the graphs, the remaining components of the hyperparameter vector $\boldsymbol{\eta}$ are fixed at their optimal values.

and a representation of the trajectory in the time-delayed embedding space of the force coefficients (plot d). In the chaotic regime of the fluidic pinball, the main peak in the PSD of C_l is notably broad. However, a predominant one is found corresponding to $St_D = 0.148$, associated with vortex shedding (with a shedding period $T_{sh} = 6.76\,c.u.$). The curve also shows a second peak associated with a lower Strouhal number ($St_D = 0.013$) due to resonance in the flow field related to the finite size of the domain, as also observed in Deng *et al.* (2020).

The condition of flow symmetry ($\bar{C}_l \approx 0$) is recovered in the chaotic regime, unlike the lower-Reynolds-number regimes which exhibit non-symmetric wakes (Deng *et al.*, 2020). In addition, C_l standard deviation is $sd(C_l) = 0.112$. As for the C_d, its free dynamics exhibits $\bar{C}_d = 3.46$ and $sd(C_d) = 0.0658$.

Figure 9 shows the controlled case in ideal measurement conditions (i.e. no noise), with the control parameters automatically selected using the BO algorithm. The simulations presented from now on include an initial unforced phase before activating the control at time instant $T_c = 50\,c.u.$. Figure 9(a, c, e) shows the time histories of the controlled aerodynamic coefficients and the exogenous input provided to the system according to the MPC algorithm. Figure 9(b, d, f) instead, displays the streamwise velocity component u averaged over time windows of one shedding period of the fluidic pinball, in correspondence of the unforced, transient and steady control phases, respectively.

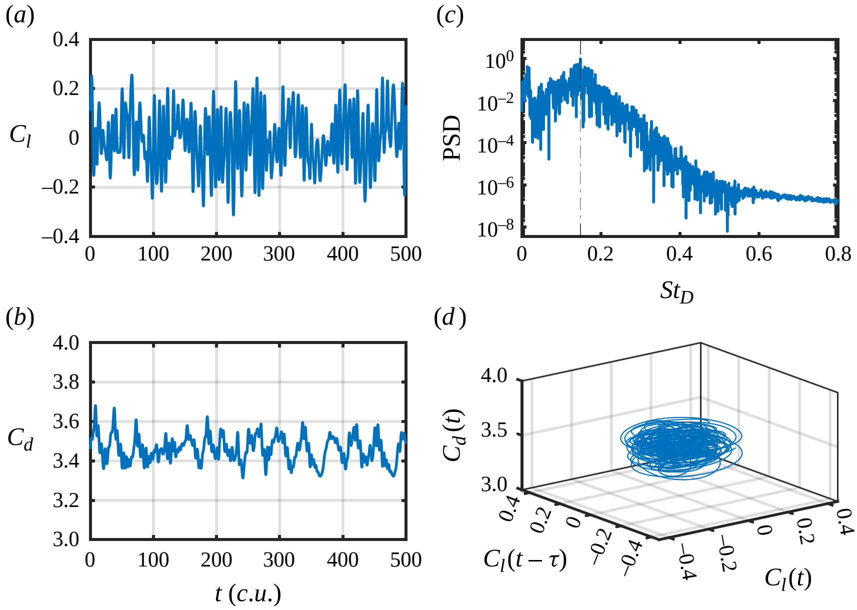

Figure 8: Characteristics of the flow field around the fluidic pinball in undisturbed conditions. Panels $(a-b)$ show the global lift and drag coefficients of the fluidic pinball. Plot (c) shows a PSD of the lift coefficient while (d) provides a representation of the trajectory in the time-delayed embedding space of the force coefficients $C_d(t)$, $C_l(t)$ and $C_l(t-\tau)$, where τ is a quarter of the shedding period of the fluidic pinball wake. The peak in the PSD is at a Strouhal number $St_D = 0.148$.

The control automatically selects an implementation law with the two rear cylinders in counter-rotation. The upper cylinder rotates clockwise ($b^2 = -b^3 < 0$) with nearly constant rotational speed equal to the maximum values set in the optimization problem. This actuation mechanism corresponds to boat tailing, which has been previously observed and studied in various works on control of the wake of a fluidic pinball (Raibaudo *et al.*, 2020; Li *et al.*, 2022).

This mechanism primarily aims to reduce the pressure drag of the fluidic pinball. Under unactuated conditions, as observed in plot A of Figure 9, an extended recirculation region with low pressure and velocity forms behind the rear cylinders. The boat tailing redirects the shear layers of the top and bottom cylinder toward the pinball axis. This streamlining process of the wake increases the pressure behind the cylinders, delays the separation and energises the wake. Furthermore, the gap jet between the cylinders is substantially weakened. A detailed description of this mechanism can be found in Geropp & Odenthal (2000). A full wake stabilization is not achieved, i.e. the vortex shedding is not suppressed, due to the imposed constraints of $\left|b^i\right| \leq 1$. This is in line with the study of Cornejo Maceda *et al.* (2021), where wake stabilization in boat tailing is achieved with rotation of the rear cylinder of 2.375 or larger.

A condition of global optimum in terms of minimizing the drag coefficient would be achieved with an actuation that involves the rear cylinders in boat tailing and the front

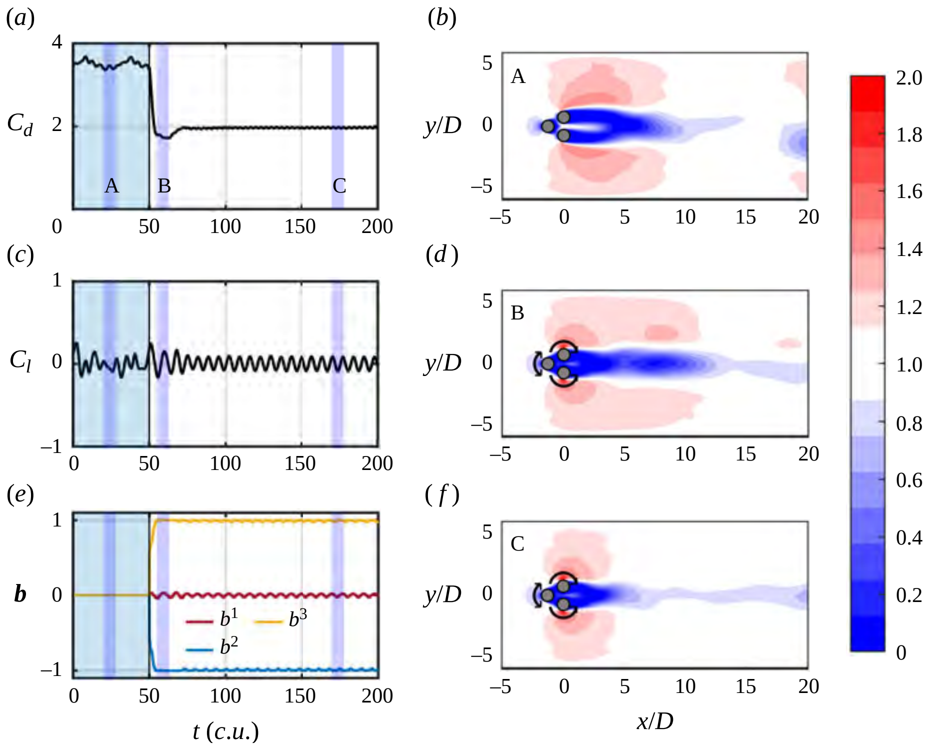

Figure 9: Results of the control application in the absence of measurement noise in the sensors. Graphics $(a-c-e)$ show the time histories of the C_d, C_l and input vector $\boldsymbol{b}$ during an initial unforced phase and then during an active control phase. Graphics $(b-d-f)$ show the mean value of the streamwise component of the velocity (u) in the time frames denoted A, B and C, respectively. Black arrows indicate the direction of rotation of the cylinders.

cylinder in a constant non-zero rotation, as observed in simulations (Li *et al.*, 2022) and also experiments (Raibaudo *et al.*, 2020). The rotation of the front cylinder helps to reduce the low-velocity areas in the region behind the cylinders of the fluidic pinball, allowing for a slight reduction in drag. However, this condition is not achieved in the present case since the control target includes a condition of null C_l, with the consequence that any asymmetric flow actuation is penalised in the optimization process. Model predictive control, on the other hand, focuses its strategy towards reducing C_l oscillations. Cornejo Maceda *et al.* (2021) document oscillating lift coefficient with amplitude increasing with increasing $b^2, -b^3$ in the range $0-2$ if solely the rear cylinders are put in rotation. The self-tuned MPC converges to an actuation strategy to reduce C_l oscillations based on phasor control with the front cylinder.

An examination of the behaviour of the drag coefficient (C_d) resulting from the implementation of the MPC control shows a decrease in its mean value, reducing it by 43.3% compared with the unforced case. Moreover, the standard deviation of the drag coefficient is reduced as well, by 85.8% during forced conditions compared with the unforced ones.

Figure 10 shows the out-of-plane vorticity in three different phases: uncontrolled, transient and post-transient the application of the control to the fluidic pinball wake. The MPC actuation is able to reduce the interaction between the upper and lower shear layers. In addition, the wake behind the fluidic pinball exhibits a more regular pattern in the post-transient phase. The wake meandering is strongly reduced, possibly due to the front stagnation point control with the front cylinder. The wake streamlining due to the boat tailing effect might also be contributing in this direction. This is also observed in the oscillating behaviour of the C_l, which becomes more regular and with less intense peaks once the control is activated. Indeed, in the controlled phase, the PSD of the C_l presents a single very narrow peak concentrated at a Strouhal number of $St_D \approx 0.14$, resulting in a shedding period of $7.16\,c.u.$, very similar to the predominant shedding frequency in the free response.

An analysis of the total power trend is required to assess if the MPC is able to achieve a net energy saving. The total power (P_{tot}) is characterised by the sum of two contributions: the first is directly related to the drag of the fluidic pinball (P_d), the second one is associated with the power required for actuation (P_a). Thus, the total power can be defined as

$$P_{tot} = P_d + P_a = F_d U_\infty + \sum_{i=1}^{3} \frac{\tau^i b^i}{R} \tag{24}$$

where τ^i is the torque acting on the i^{th} cylinder of the fluidic pinball. The time history of power during the free and forced phases is given in Figure 11. It can be observed that under undisturbed conditions, the total power remains around an average value of 1.75, with oscillations related only to the drag of the fluidic pinball. When moving to the controlled solution, the total power undergoes a sharp decrease in the transient phase thanks to the boat tailing. The total power stabilises at about 1.12, experiencing a reduction of 36.31% compared with the unforced phase. It is also observed that the average actuation power corresponds only to 12.13% of the total power in the controlled phase. The average value of the actuation power is mainly related to the rear cylinders in boat tailing, while its oscillations are due to the front cylinder aiming to control the shedding, in order to reduce the lift fluctuations. It must be remarked though that this does not take into account possible inefficiencies that would arise in an implementation in a real environment.

A comparison between the proper orthogonal decomposition (POD) of the flow fields with and without control according to the MPC algorithm is also shown in Figure 12. It is specified that the POD has been performed on a dataset including both the transient and the stationary part of the control. On the left side of the figure, the spatial distributions of the first, third, and fifth out-of-plane vorticity modes are shown for both free and controlled cases. The plots on the right side of the figure show the eigenvalues of the first 100 temporal modes. The eigenvalues are normalised with respect to their sum for the case without actuation. This allows a direct comparison of the mode energy in unforced and forced cases.

By observing the graphs related to the POD of the wake dataset under controlled conditions, it is noted that shedding inception is shifted upstream if compared with the unforced condition and is also more regular in the presence of boat tailing of the rear

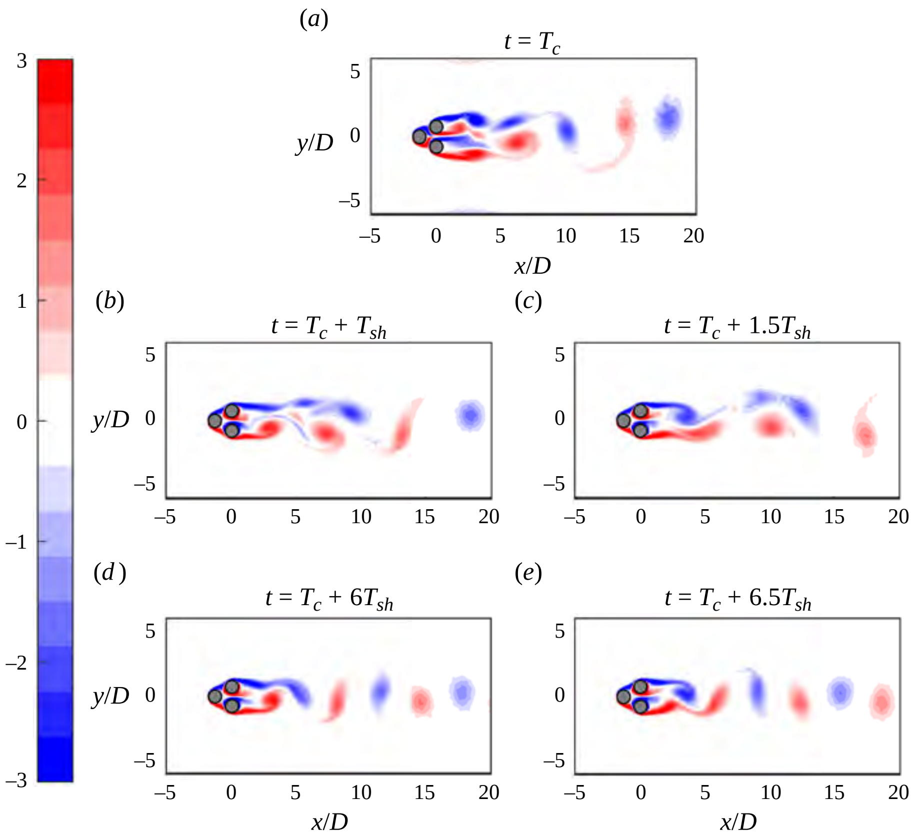

Figure 10: Contour plot of the out-of-plane vorticity component of the flow around the fluidic pinball. Several instants corresponding to the phases of control onset, transient and post-transient phases are presented. Here T_{sh} refers to the characteristic shedding period of the fluidic pinball in unforced conditions.

cylinders. Furthermore, the modes of the controlled case, compared with the free case, are characterised by lower energy, with the exception of the first two modes. This is ascribed to a partial wake stabilization, with reduced meandering. Mode 1 spotlights indeed more defined vortical structures, with the crosswise width practically constant throughout the wake development. The fluidic boat tailing of the wake has indeed the effect of re-energizing the outer shear layers, redirecting them towards the pinball axis. The wake dynamics shift towards a less chaotic behaviour, as also observed in the more regular oscillations of the C_l after activation of the control (Figure 9). A less rich dynamics results in a more compact POD eigenspectrum for the controlled case, with higher energy in the first two modes. The reduced crosswise oscillations are observed also in the structure of higher-order modes. Modes $3-6$ for the uncontrolled case embed the energy of higher-order harmonics and model the crosswise wake oscillations. In addition to the fluidic boat tailing due to the rotation of the rear cylinders, the stagnation point control enforced by the front cylinder has the effect of weakening such oscillations. This results in lower energy for the

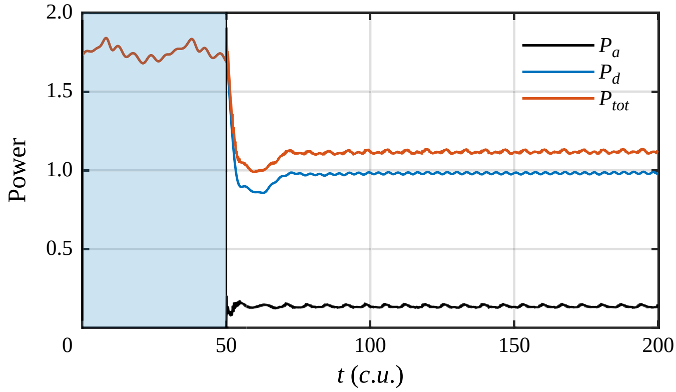

Figure 11: Time histories of the power associated with drag, actuation and total power during the free and forced stages. Simulation of the fluidic pinball wake control in the absence of measurement noise.

corresponding POD modes in the controlled case, and a more compact structure of their vortical features.

3.3.2. *The effects of measurements noise in the sensors*

The self-tuned control strategy with online LPR of the sensor signal is assessed in the presence of noise. The control results in terms of C_d and C_l and actuation are given in Figure 13. We include as a reference the case of MPC without hyperparameter tuning, i.e. the control parameters are manually selected and all set to unity (except for the prediction/control window set to $3c.u.$). The outcomes when the parameter selection is automatically performed using BO is reported for different noise levels. Without hyperparameter tuning, the performance of the control is quite poor for both C_d and C_l. In this case, the mean drag coefficient is 3.0598, i.e. significantly larger than the value of ≈ 1.97 achieved after hyperparameter tuning. Furthermore, MPC converges to an asymmetric controlled state, with a negative average lift coefficient. Quite surprisingly, this is achieved even if $R_{b^2} = R_{b^3} = 1$, i.e. forcing symmetry in the hyperparameters does not necessarily lead to symmetric actuation. After hyperparameter tuning, the penalty weight of actuation (and actuation variability) on the rear cylinder is decreased with respect to the front cylinder (see Table 1). This allows stronger actuation on the rear cylinders, thus fostering strong fluidic boat tailing, with consequent strong drag reduction, accompanied by a weak action of the front cylinder to improve wake stability.

The cases analysed above are also reproduced in Figure 14 in terms of trajectories of $C_d(t)$, $C_l(t)$ and $C_l(t-\tau)$, where τ corresponds to a quarter of the shedding period T_{sh} of the fluidic pinball wake. The plots offer a description of the attractor onto which the wake dynamics of the fluidic pinball evolves, including the transitional phase from free to forced conditions. When the parameter tuning is performed, during the control C_d experiences a remarkable reduction, whereas C_l evolves on a dynamics similar to the unforced case, although less chaotic. Nevertheless, it is worth noting that the control of C_l is not perfectly attained, as the model uncertainties hinder a complete description of

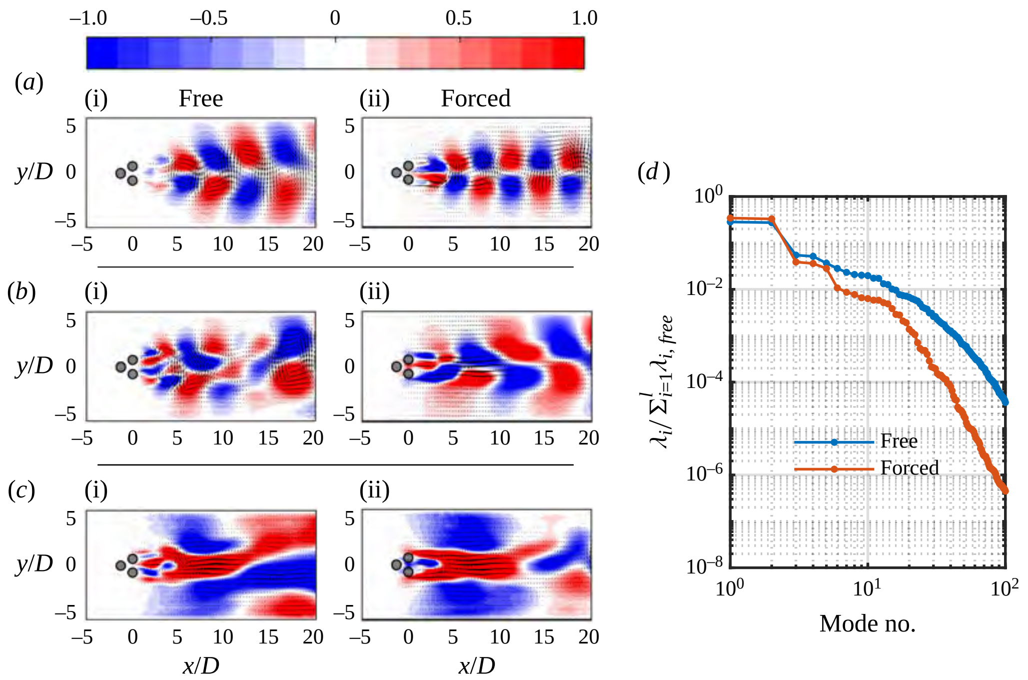

Figure 12: Results of a POD applied to the flow field of the wake past fluidic pinball. Panels $(a-c)$, from top to bottom depict the representation of out-of-plane vorticity for the spatial modes 1, 3 and 5 under both free and forced conditions in accordance with the MPC architecture. Every plot also presents velocity vectors corresponding to each spatial mode. Plot (d) the squared singular values (λ_i) of the POD, normalized with respect to the sum of the squared singular values for the free case $(\lambda_{i,free})$. Proper orthogonal decomposition performed on a dataset consisting of $l = 1200$ snapshots of the fluidic pinball wake, including the transient in the forced case.

the shedding dynamics of the fluidic pinball. Without hyperparameter tuning, in the first part of the transient the system transitions initially towards a state with a lower average C_d (around 2.5) and C_l oscillations of slightly lower intensity. The relatively high penalty on the actuation of the rear cylinders with respect to the front one, however, forces the system to avoid strong boat tailing and starts leveraging asymmetric actuation based on the front cylinder in the attempt to stabilise the wake through stagnation point control. The system finally converges to a limit cycle with a higher average C_d and with large oscillations of the C_l around a negative average.

In terms of the C_d, the control remains nearly unaffected by variations in measurement noise in the sensors. Regardless of the noise intensity, the posterior cylinders consistently maintain their boat tailing configuration, exhibiting maximum rotational velocity limited by the actuation constraints. On the other hand, the control of the lift coefficient demonstrates higher sensitivity to measurement noise. This is attributed to the increased susceptibility of the predictive model to disturbances in the initial conditions of online

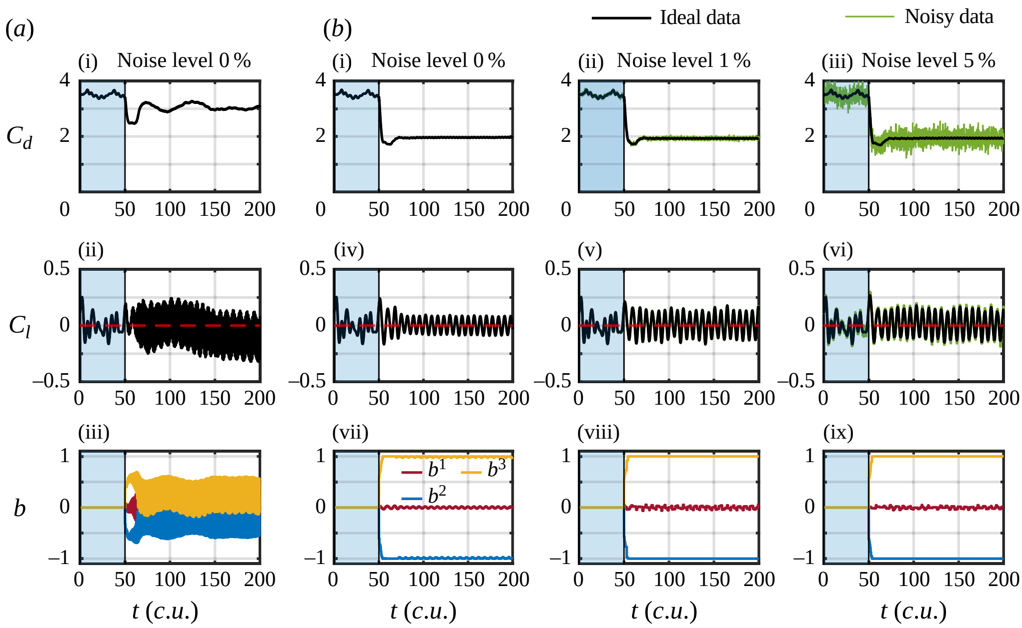

Figure 13: Time series of C_d, C_l and exogenous input b^i (from top to bottom row, respectively) at free and forced conditions according to the MPC framework applied to the fluidic pinball. The panels in (a) show the results when the cost function parameters are hand selected and equal to the unity (with the exception of the prediction control window, set to $3c.u.$). The panels in (b) show the results when BO is used for hyperparameter tuning. Local polynomial regression is included in the MPC framework in all cases where measurement noise is present.

predictions. As the measurement noise intensifies, the frontal cylinder displays larger oscillations in the selected actuation law prescribed by the MPC.

Nevertheless, LPR provides advantages to the algorithm, resulting in only marginal degradation of control performance. This increases the robustness of the application in realistic control scenarios characterised by high sensor noise.

3.3.3. *Symmetry in rear cylinder parameters*

This part is intended to provide more details about the choice of symmetry in the control parameters. Specifically, it concerns the components of the weight matrices that penalise the input expenditures and input variability of the rear cylinders during control, already introduced in § 2.1.4.

The results of the control simulations with and without imposed symmetry conditions are presented in Figure 15. A certain bias is observed in the control of C_l in the case without imposing symmetry conditions. This is related to an asymmetry in the rotation of the rear cylinders. However, under imposed symmetry, the control of the drag coefficient improves slightly, as the tangential speeds of rotation of the rear cylinders are both at the actuation limits, creating a stronger boat tailing. This lower effectiveness of the

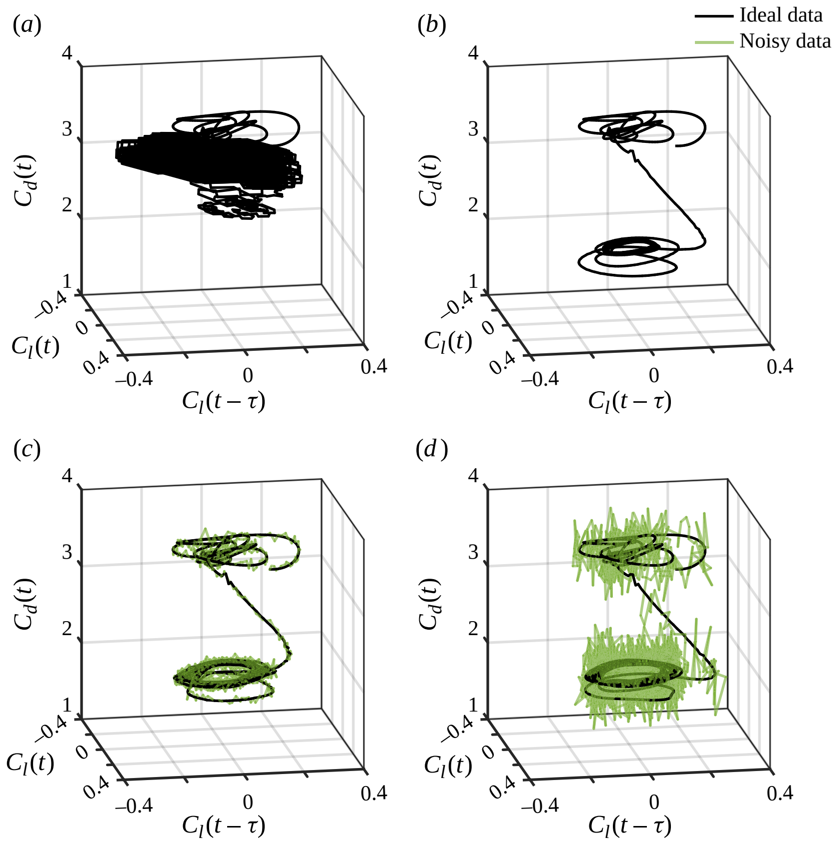

Figure 14: Trajectories in the time-delayed embedding space of the force coefficients $C_d(t)$, $C_l(t)$ and $C_l(t-\tau)$, where τ is a quarter of the shedding period of the fluidic pinball wake. The panel in (a) shows the results when the cost function parameters are hand selected and equal to unity, while panels $(b-d)$ show the result when BO is used for hyperparameter tuning. Cases $(a-b)$ relate to ideal measurement conditions while cases $(c-d)$ have noise of 1% and 5%, respectively. Local polynomial is included in the MPC control framework in all cases where measurement noise is present.

optimization in the case without imposed symmetry can be explained by the relatively low importance of the C_l fluctuations in the functional of BO in equation (18), as discussed in § 3.2. The implementation of a law that binds the parameters of the rear cylinders reduces the total number of parameters to be tuned, leading to a less computationally expensive offline phase of the control.

4. Discussion and conclusion

In this work a noise-robust MPC algorithm that does not require user intervention neither for the plant modelling nor for the hyperparameters selection is proposed. The model used in the MPC optimization is selected from input-output data of the system under control

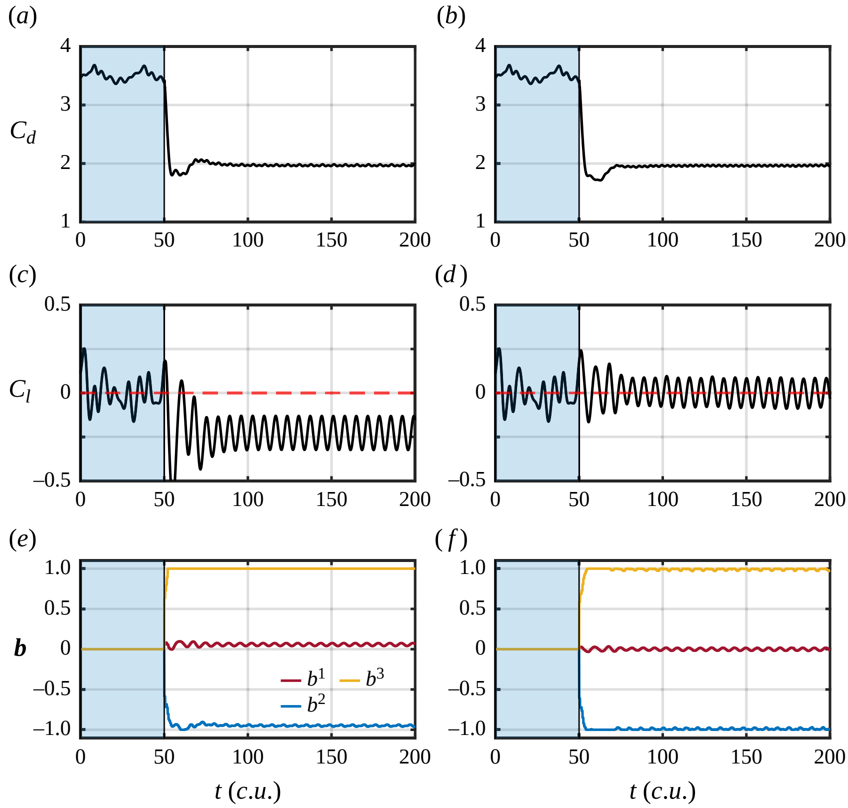

Figure 15: Results of the application of MPC in terms of C_d, C_l and input vector $\boldsymbol{b}$, without $(a-c-e)$ and with $(b-d-f)$ the imposition of symmetry in the parameters related to the rear cylinders. In both cases, all the parameters are tuned using the BO algorithm. Results of control in the absence of measurement noise in the sensors.

using nonlinear system identification. The control parameters are automatically selected using a black-box optimization based on Bayesian methods. Additionally, the robustness of the algorithm is enhanced by the non-parametric smoothing technique LPR, which acts on the output data from the control sensors to address the presence of measurement noise.

The proposed control algorithm is successfully applied to the control of the wake of the fluidic pinball for drag reduction in a chaotic regime (specifically, for $Re_D = 150$) using solely aerodynamic forces to guide the control strategy. The methodology achieves good success in controlling a non-linear, chaotic and high-dimensional system. Being based on the MPC technique, the algorithm easily allows for the inclusion of non-linear constraints in the control and is very promising for applications where the control target is not a simple set-point stabilization.

The proposed technique has shown to be highly robust to sensor measurement uncertainty, performing excellently even in realistic control scenarios characterised by a high level of noise. It must be remarked that, although rather realistic for force measurements, the Gaussian noise model might not be adequate if other sensing techniques are used. This may have an impact on plant model identification and LPR performance. However, the most attractive feature is that the method requires minimal user interaction, as the control parameters are automatically tuned by the BO algorithm according to the control target, taking into account also different fidelity levels for the plant predictions. The automatic procedure identified the most rewarding directions to optimise parameters with the aid only of a few numerical experiments. To control C_d and C_l of the fluidic pinball around a zero set point the MPC algorithm used a combination of two control mechanisms that have been previously considered in fluidic pinball control works: the boat tailing of the rear cylinders for drag reduction and the stagnation point control of the front cylinder for shedding control. This corresponded to a stronger penalty on the C_d and a low penalty on the actuation cost of the rear cylinders. Without the framework proposed in this work, this process would have required parametric studies or suboptimal analysis, with the inherent difficulty of choosing relative weights between heterogeneous quantities.

It is worth highlighting that the control algorithm is currently being tested on a control case with relatively simple dynamics (Deng *et al.*, 2020, 2022). This streamlines the process of coordinate selection and nonlinear system identification using the SINDYc technique. On the one hand, the aerodynamic force coefficients in separated wake flows exhibit features that render them suitable for plant modelling. On the other hand, the dynamics of the flow, although chaotic, evolve clearly on a low-dimensional attractor, thus simplifying the library selection for the application of SINDYc. For more complex flows (higher Reynolds number or with less clear features for straightforward coordinate identification), we envision that the challenge of MPC application will reside mostly in the coordinate selection and in the plant identification. Fast-paced advancements in grey- and black-box modelling are paving the way to interesting research pathways in these directions. The approach proposed here, however, will still be applicable and will benefit from such advancements. The offline optimization of the hyperparameters of the cost function can be applied independently of the method for coordinate selection or plant identification. Furthermore, the outcome will automatically adapt to the fidelity of the model plant when compared with the uncertainty of the measurements used to feed it.

References

Ahuja, S., Rowley, C., Kevrekidis, I., Wei, M., Colonius, T. & Tadmor, G. 2007 Low-dimensional models for control of leading-edge vortices: equilibria and linearized models. In *45th AIAA Aerospace Sciences Meeting and Exhibit*, p. 709.

Allgöwer, F., Findeisen, R. & Nagy, Z. K. 2004 Nonlinear model predictive control: From theory to application. *J. Chin. Inst. Chem. Eng.* **35** (3), 299–316.

Beintema, G., Corbetta, A., Biferale, L. & Toschi, F. 2020 Controlling rayleigh–bénard convection via reinforcement learning. *J. Turbul.* **21** (9-10), 585–605, arXiv: https://doi.org/10.1080/14685248.2020.1797059.

Bewley, T. R., Moin, P. & Temam, R. 2001 Dns-based predictive control of turbulence: an optimal benchmark for feedback algorithms. *J. Fluid Mech.* **447**, 179–225.

Bieker, K., Peitz, S., Brunton, S. L., Kutz, J. N. & Dellnitz, M. 2020 Deep model predictive control with online learning for complex physical systems. *Theoretical and Computational Fluid Dynamics* **34** (10).

Blackburn, H. M. & Henderson, R. D. 1999 A study of two-dimensional flow past an oscillating cylinder. *J. Fluid Mech.* **385**, 255–286.

Bøhn, E., Gros, S., Moe, S. & Johansen, T. A. 2023 Optimization of the model predictive control meta-parameters through reinforcement learning. *Eng. Appl. Artif. Intell.* **123**, 106211.

Brunton, S. L. & Noack, B. R. 2015 Closed-loop turbulence control: Progress and challenges. *Appl. Mech. Rev.* **67** (5), 050801.

Brunton, S. L., Proctor, J. L. & Kutz, J. N. 2016*a* Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proc. Natl. Acad. Sci. U.S.A.* **113** (15), 3932–3937.

Brunton, S. L., Proctor, J. L. & Kutz, J. N. 2016*b* Sparse identification of nonlinear dynamics with control (sindyc). *IFAC-PapersOnLine* **49** (18), 710–715, 10th IFAC Symposium on Nonlinear Control Systems NOLCOS 2016.

Buche, D., Stoll, P., Dornberger, R. & Koumoutsakos, P. 2002 Multiobjective evolutionary algorithm for the optimization of noisy combustion processes. *IEEE Trans. Syst. Man Cybern.* **32** (4), 460–473.

Camacho, E. F. & Alba, C. B. 2013 *Model predictive control*. Springer science & business media.

Castellanos, R., Cornejo Maceda, G. Y., de la Fuente, I., Noack, B. R., Ianiro, A. & Discetti, S. 2022*a* Machine-learning flow control with few sensor feedback and measurement noise. *Phys. Fluids* **34** (4), 047118.

Castellanos, R., Michelis, T., Discetti, S., Ianiro, A. & Kotsonis, M. 2022*b* Reducing turbulent convective heat transfer with streamwise plasma vortex generators. *Exp. Therm. Fluid Sci.* **134**, 110596.

Cetiner, O. & Rockwell, D. 2001 Streamwise oscillations of a cylinder in a steady current. part 1. locked-on states of vortex formation and loading. *J. Fluid Mech.* **427**, 1–28.

Chevalier, M., Hœpffner, J., Åkervik, E. & Henningson, D. S. 2007 Linear feedback control and estimation applied to instabilities in spatially developing boundary layers. *J. Fluid Mech.* **588**, 163–187.

Collis, S., Chang, Y., Kellogg, S. & Prabhu, R. 2000 Large eddy simulation and turbulence control. In *Fluids 2000 Conference and Exhibit*, p. 2564.

Cornejo Maceda, G. Y., Li, Y., Lusseyran, F., Morzyński, M. & Noack, B. R. 2021 Stabilization of the fluidic pinball with gradient-enriched machine learning control. *J. Fluid Mech.* **917**, A42.

Corona, D. & De Schutter, B. 2008 Adaptive cruise control for a smart car: A comparison benchmark for mpc-pwa control methods. *IEEE Trans. Control Syst. Technol.* **16** (2), 365–372.

Cortelezzi, L. & Speyer, J. L. 1998 Robust reduced-order controller of laminar boundary layer transitions. *Phys. Rev. E.* **58** (2), 1906.

Deng, N., Noack, B. R., Morzyński, M. & Pastur, L. R. 2020 Low-order model for successive bifurcations of the fluidic pinball. *J. Fluid Mech.* **884**, A37.

Deng, N., Noack, B. R., Morzyński, M. & Pastur, L. R. 2022 Cluster-based hierarchical network model of the fluidic pinball–cartographing transient and post-transient, multi-frequency, multi-attractor behaviour. *J. Fluid Mech.* **934**, A24.

Duriez, T., Brunton, S. L. & Noack, B. R. 2017 *Machine learning control-taming nonlinear dynamics and turbulence*, , vol. 116. Springer.

Edwards, W., Tang, G., Mamakoukas, G., Murphey, T. & Hauser, K. 2021 Automatic tuning for data-driven model predictive control. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7379–7385. IEEE.

Fan, D., Yang, L., Wang, Z., Triantafyllou, M. S. & Karniadakis, G. E. 2020 Reinforcement learning for bluff body active flow control in experiments and simulations. *Proc. Natl. Acad. Sci. U.S.A.* **117** (42), 26091–26098.

Fan, J. & Gijbels, I. 1996 *Local polynomial modelling and its applications*. *Monographs on statistics and applied probability series* 66. London [u.a.]: Chapman & Hall.

Fröhlich, L. P., Küttel, C., Arcari, E., Hewing, L., Zeilinger, M. N. & Carron, A. 2022 Contextual tuning of model predictive control for autonomous racing. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 10555–10562. IEEE.

Gautier, N., Aider, J.-L., Duriez, T., Noack, B. R., Segond, M. & Abel, M. 2015 Closed-loop separation control using machine-learning. *J. Fluid Mech.* **770**, 442–457.

Gerhard, J., Pastoor, M., K., R., Noack, B., Dillmann, A., Morzynski, M. & Tadmor, G. 2003 Model-based control of vortex shedding using low-dimensional galerkin models. In *33rd AIAA fluid dynamics conference and exhibit*, p. 4262.

Geropp, D. & Odenthal, H.-J. 2000 Drag reduction of motor vehicles by active flow control using the coanda effect. *Exp. Fluids* **28** (1), 74–85.

Giordano, F. & Parrella, M. L. 2008 Neural networks for bandwidth selection in local linear regression of time series. *Comput. Stat. Data Anal.* **52** (5), 2435–2450.

Green, J. E. 2003 Civil aviation and the environmental challenge. *Aeronaut. J.* **107** (1072), 281–299.

Grüne, L. & Pannek, J. 2017 Nonlinear model predictive control. In *Nonlinear model predictive control*. Springer Cham.

Gad-el Hak, M. 2000 *Flow Control: Passive, Active, and Reactive Flow Management*. Cambridge University Press.

Henson, M. A. 1998 Nonlinear model predictive control: current status and future directions. *Comput. Chem. Eng.* **23** (2), 187–202.

Hewing, L., Wabersich, K. P., Menner, M. & Zeilinger, M. N. 2020 Learning-based model predictive control: Toward safe learning in control. *Annu. Rev. Control Robot. Auton. Syst.* **3** (1), 269–296.

Illingworth, S. J., Morgans, A. S. & Rowley, C. W. 2011 Feedback control of flow resonances using balanced reduced-order models. *J. Sound Vib.* **330** (8), 1567–1581.

Kaiser, E., Kutz, J. N. & Brunton, S. L. 2018 Sparse identification of nonlinear dynamics for model predictive control in the low-data limit. *Proc. R. Soc. Lond.* **474** (2219), 20180335.

Kaiser, E., Noack, B. R., Spohn, A., Cattafesta, L. N. & Morzyński, M. 2017 Cluster-based control of a separating flow over a smoothly contoured ramp. *Theor. Comput. Fluid Dyn.* **31**, 579–593.

Kim, J. 2011 Physics and control of wall turbulence for drag reduction. *Philos. Trans. Royal Soc. A* **369** (1940), 1396–1411.

Kim, J. & Bewley, T. R. 2007 A linear systems approach to flow control. *Annu. Rev. Fluid Mech.* **39** (1), 383–417.

Koumoutsakos, P., Freund, J. & Parekh, D. 2001 Evolution strategies for automatic optimization of jet mixing. *AIAA J.* **39** (5), 967–969, arXiv: https://doi.org/10.2514/2.1404.

Lee, J. H. 2011 Model predictive control: Review of the three decades of development. *Int. J. Control. Autom.* **9** (3), 415–424.

Lee, K. H., Cortelezzi, L., Kim, J. & Speyer, J. 2001 Application of reduced-order controller to turbulent flows for drag reduction. *Phys. Fluids* **13** (5), 1321–1330.

Li, Y., Cui, W., Jia, Q., Li, Q., Yang, Z., Morzyński, M. & Noack, B. R. 2022 Explorative gradient method for active drag reduction of the fluidic pinball and slanted ahmed body. *J. Fluid Mech.* **932**, A7.

Little, J., Debiasi, M., Caraballo, E. & Samimy, M. 2007 Effects of open-loop and closed-loop control on subsonic cavity flows. *Phys. Fluids* **19** (6), 065104.

Loiseau, J. C., Noack, B. R. & Brunton, S. L. 2018 Sparse reduced-order modelling: sensor-based dynamics to full-state estimation. *J. Fluid Mech.* **844**, 459–490.

Monokrousos, A., Brandt, L., Schlatter, P. & Henningson, D. S. 2008 Dns and les of estimation and control of transition in boundary layers subject to free-stream turbulence. *Int. J. Heat Fluid Flow* **29** (3), 841–855.

Morton, J., Jameson, A., Kochenderfer, M. J. & Witherden, F. 2018 Deep dynamical modeling and control of unsteady fluid flows. In *Advances in Neural Information Processing Systems* (ed. S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi & R. Garnett), , vol. 31. Curran Associates, Inc.

Nagarajan, K. K., Singha, S., Cordier, L. & Airiau, C. 2018 Open-loop control of cavity noise using proper orthogonal decomposition reduced-order model. *Comput Fluids* **160**, 1–13.

Nair, A. G., Yeh, Chi-An, Kaiser, E., Noack, B. R., Brunton, S. L. & Taira, K. 2019 Cluster-based feedback control of turbulent post-stall separated flows. *J. Fluid Mech.* **875**, 345–375.

Noack, B. R. & Morzyński, M. 2017 The fluidic pinball—a toolkit for multiple-input multiple-output flow control (version 1.0). *Tech. Rep.*. Tech. Rep. 02/2017. Chair of Virtual Engineering, Poznan University of Technology.

Ouyang, L., Zhou, D., Ma, Y. & Tu, Y. 2018 Ensemble modeling based on 0–1 programming in micro-manufacturing process. *Comput. Ind. Eng.* **123**, 242–253.

Parkin, D. J., Thompson, M. C. & Sheridan, J. 2014 Numerical analysis of bluff body wakes under periodic open-loop control. *J. Fluid Mech.* **739**, 94–123.

Pastur, L. R., Deng, N., Morzyński, M. & Noack, B. R. 2019 Reduced-order modeling of the fluidic pinball. In *Chaotic Modeling and Simulation International Conference*, pp. 205–213. Springer, Cham: Springer International Publishing.

Peitz, S., Otto, S. E. & Rowley, C. W. 2020 Data-driven model predictive control using interpolated koopman generators. *SIAM J. Appl. Dyn. Syst.* **19** (3), 2162–2193.

Pinier, J. T., Ausseur, J. M., Glauser, M. N. & Higuchi, H. 2007 Proportional closed-loop feedback control of flow separation. *AIAA J.* **45** (1), 181–190.

Poncet, P., Cottet, G. H. & Koumoutsakos, P. 2005 Control of three-dimensional wakes using evolution strategies. *CR Mécanique* **333** (1), 65–77, high-order methods for the numerical simulation of vortical and turbulent flows [special issue].

Qin, S. J. & Badgwell, T. A. 2003 A survey of industrial model predictive control technology. *Control Eng. Pract.* **11** (7), 733–764.

Rabault, J., Kuchta, M., Jensen, A., Réglade, U. & Cerardi, N. 2019 Artificial neural networks trained through deep reinforcement learning discover control strategies for active flow control. *J. Fluid Mech.* **865**, 281–302.

Raibaudo, C., Zhong, P., Noack, B. R. & Martinuzzi, R. J. 2020 Machine learning strategies applied to the control of a fluidic pinball. *Phys. Fluids* **32** (1), 015108.

Raković, S. V. & Levine, W. S. 2018 *Handbook of Model Predictive Control*. *Control engineering* 1. Cham: Springer International Publishing AG.

Rowley, C. W. & Williams, D. R. 2006 Dynamics and control of high-reynolds-number flow over open cavities. *Annu. Rev. Fluid Mech.* **38**, 251–276.

Sasaki, Y. & Tsubakino, D. 2020 Designs of feedback controllers for fluid flows based on model predictive control and regression analysis. *Energies* **13** (6), 1325.

Schumm, M., Berger, E. & Monkewitz, P. A. 1994 Self-excited oscillations in the wake of two-dimensional bluff bodies and their control. *J. Fluid Mech.* **271**, 17–53.

Shimomura, S., Sekimoto, S., Oyama, A., Fujii, K. & Nishida, H. 2020 Closed-loop flow separation control using the deep q network over airfoil. *AIAA J.* **58** (10), 4260–4270.

Sipp, D. 2012 Open-loop control of cavity oscillations with harmonic forcings. *J. Fluid Mech.* **708**, 439–468.

Snoek, J., Larochelle, H. & Adams, R. 2012 Practical bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems* (ed. F. Pereira, C. J. Burges, L. Bottou & K. Q. Weinberger), , vol. 25. Curran Associates, Inc.

Steffen, J., Oztop, E. & Ritter, H. 2010 Structured unsupervised kernel regression for closed-loop motion control. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 75–80.

Tadmor, G., Lehmann, O., Noack, B. R., Cordier, L., Delville, J., Bonnet, J. P. & Morzyński, M. 2011 Reduced-order models for closed-loop wake control. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* **369** (1940), 1513–1524.

Thiria, B., Goujon-Durand, S. & Wesfreid, J. E. 2006 The wake of a cylinder performing rotary oscillations. *J. Fluid Mech.* **560**, 123–147.

Tol, H. J., De Visser, C. C. & Kotsonis, M. 2019 Experimental model-based estimation and control of natural tollmien–schlichting waves. *AIAA J.* **57** (6), 2344–2355.

Wand, M. P. & Jones, M. C. 1994 *Kernel Smoothing. Chapman & Hall/CRC Monographs on Statistics & Applied Probability* 60. Boca Raton, FL, U.S.: Chapman & Hall.

Wang, X., Deng, N., Cornejo Maceda, G. Y. & Noack, B. R. 2023 Cluster-based control for net drag reduction of the fluidic pinball. *Phys. Fluids* **35** (2), 023601.

Wu, Z., Fan, D., Zhou, Y., Li, R. & Noack, B. R. 2018 Jet mixing optimization using machine learning control. *Exp. Fluids* **59**, 1–17.

Paper 2

2

Latent-space non-linear model predictive control for partially-observable systems

L. Marra[1], O. Semeraro[2], L. Mathelin[2], A. Meilán-Vila[3] & S. Discetti[1]

[1] Department of Aerospace Engineering, Universidad Carlos III de Madrid, Av. de la Universidad 30, Leganés, 28911, Madrid, Spain
[2] CNRS, Laboratoire interdisciplinaire des sciences du numérique (LISN), Université Paris-Saclay, 91400 Orsay, France
[3] Department of Statistics, Universidad Carlos III de Madrid, Calle Madrid, 126, Getafe, 28903, Madrid, Spain

Under review (2025), DOI:
`https://doi.org/10.48550/arXiv.2511.19056`

This work presents a scalable control framework based on nonlinear Model Predictive Control for high-dimensional dynamical systems. The proposed approach addresses the key challenges of model scalability and partial observability by integrating data-driven reduced order modelling, control in a latent space, and state estimation within a unified formulation. A predictive model is constructed via Operator Inference on a Proper Orthogonal Decomposition basis, yielding a compact latent representation that captures the dominant system dynamics. State estimation is achieved through an Unscented Kalman Filter, which reconstructs the latent space from sparse and noisy measurements, enabling closed-loop control. The input signals are computed directly in the reduced-order latent space, improving computational efficiency with negligible impact on predictive capability. The methodology is validated on the one- and two-dimensional Kuramoto–Sivashinsky equations, serving as benchmarks for chaotic and spatially-extended systems. Numerical experiments demonstrate that the proposed framework achieves accurate stabilisation. Overall, the framework provides a practical approach for nonlinear control of complex, high-dimensional systems where full-state measurements are often inaccessible or infeasible.

Key words: Model Predictive Control, Operator Inference, Unscented Kalman Filter

---

1. Introduction

Controlling dynamical systems arising in scientific, engineering, and industrial applications can pose significant challenges, as these systems may exhibit complex behaviours. Such systems often exhibit high dimensionality, multiscale spatio-temporal dynamics, strong nonlinearities, time delays, or chaotic behaviour (Brunton & Kutz, 2022). These characteristics make the design of efficient real-time control strategies both computationally demanding and, in many cases, practically infeasible. Such difficulties are particularly evident in fluid dynamics, where flows are typically two- or three-dimensional, governed by nonlinear equations, and characterised by intricate multiscale structure (Brunton & Noack, 2015).

A common strategy is model-based control, in which a possibly surrogate model of the system dynamics is used to determine control actions. These approaches, implicitly or explicitly, incorporate knowledge of the system, sometimes using a predictive model to simulate the response under different inputs. However, for complex nonlinear systems, accurate long-term prediction is rarely possible. Sensitivity to initial conditions, unmodeled dynamics, and measurement noise typically restrict reliable predictions to short horizons.

Model Predictive Control (MPC, Camacho & Bordons, 2007) is a powerful control framework that uses predictive models over a finite horizon to determine control actions. At each time step, MPC solves an optimisation problem of a prescribed cost functional, which encodes the control objectives. This produces a sequence of future control actions, of which only the first is implemented. The horizon then shifts forward, and the optimisation is repeated using updated system information. This strategy, known as receding-horizon control, gives MPC both flexibility and robustness, enabling its application to linear (Muske & Rawlings, 1993) and nonlinear systems (Allgower *et al.*, 2004), as well as to continuous and discrete-time formulations. By directly optimising over model predictions, MPC also allows for the incorporation of input and state constraints, making it suitable for safety-critical applications even within fluid dynamics. In this field, MPC has been successfully applied to a variety of problems, such as wall-bounded flows (Bewley *et al.*, 2001), cavity flows (Arbabi *et al.*, 2018), and wake flows such as those past circular cylinders (Morton *et al.*, 2018; Déda *et al.*, 2023) and the fluidic pinball (Marra *et al.*, 2024; Bieker *et al.*, 2020). A comparison of MPC with other model-based techniques for flow control is reported in Fabbiane *et al.* (2014). The cornerstone of the application of MPC is the availability of a reliable predictive model of the system dynamics that is accurate, computationally efficient (potentially parsimonious), robust, and whose states are directly measurable. However, for high-dimensional nonlinear dynamical systems, achieving all these properties poses a major challenge.

Regarding model construction, the choice of the approach depends on the availability of governing equations and data. In many practical settings, the equations are known and high-fidelity simulations are available, but their computational cost makes them unsuitable for direct use in control. In such cases, reduced-order modelling (ROM) provides an efficient way to approximate the essential system dynamics. These methods employ a proxy for the full system state, significantly improving computational efficiency in control. Projection-based approaches, for example, identify a low-dimensional subspace from data and project the governing equations onto it to obtain reduced dynamics, as in classical Proper Orthogonal Decomposition (POD)–Galerkin methods (Rempfer, 2000). When only measurements are available, data-driven identification methods are often used (Brunton *et al.*, 2020; Raissi & Karniadakis, 2018). Traditional parametric input–output models, such as autoregressive models (Box *et al.*, 2015; Billings, 2013), or output-error formulations (Ljung, 2003), remain popular due to their interpretability and computational efficiency. Gaussian process regression has also been employed for input-output maps (Seeger, 2004) with uncertainty quantification. Recent advances have introduced sparsity-promoting approaches such as the System Identification of Nonlinear Dynamics (SINDy, Brunton *et al.*, 2016*a*,*b*) and neural-network-based architectures (Chiuso & Pillonetto, 2019; Schüssler *et al.*, 2019), which enable the identification of complex nonlinear behaviours directly from data. Other notable examples include approaches based on reservoir computing (Özalp *et al.*, 2026; Ozan *et al.*, 2025; Nóvoa & Magri, 2025) and Long Short-Term Memory networks for prediction (Özalp *et al.*, 2023; Sangiorgio *et al.*, 2021). In parallel, Operator

Inference (OpInf, Kramer *et al.*, 2024; Peherstorfer & Willcox, 2016) has extended this idea in a data-driven, non-intrusive manner by identifying reduced operators directly from simulation or experimental data. Operator Inference has demonstrated success in learning reduced-order models even for complex partial differential equations (Qian *et al.*, 2022) and experimental test cases (Kramer *et al.*, 2024). More recently, autoencoders have also been used to map the system to a nonlinear latent space for learning dynamics (Agostini, 2020; Zhang *et al.*, 2025; Racca *et al.*, 2023; Özalp *et al.*, 2026). While they can capture a strongly nonlinear behaviour, the nonlinear mapping is often complex and less interpretable than a linear projection. ROM techniques have been increasingly integrated into MPC frameworks to achieve real-time feasibility. Examples include MPC formulations based on DMD (Lu & Zavala, 2021), Spectral Submanifolds (SSM, Alora *et al.*, 2023), and POD-based models (Hovland *et al.*, 2006; Karg & Lucia, 2021).

However, even with reduced models, MPC of high-dimensional and nonlinear systems remains limited by model inaccuracies, truncation errors, and measurement sparsity. In most practical implementations, only sparse and noisy measurements are available, necessitating an estimation mechanism to reconstruct the full or latent system state. This leads to output-feedback MPC formulations (Findeisen *et al.*, 2003), in which control decisions are based on estimated rather than directly observed states. Common estimation strategies include the Kalman Filter and its nonlinear or ensemble-based extensions, such as Extended (Ribeiro, 2004), Ensemble (EnKF, Evensen, 2003), and Unscented (UKF) Kalman Filters (Wan & Van Der Merwe, 2000), or moving horizon estimation strategies (Tenny & Rawlings, 2002). An overview of output-feedback MPC is provided in Findeisen *et al.* (2003), while examples of applications using a Kalman Filter or moving horizon estimation are included in Lee & Ricker (1994); Janiszewski (2024); Tenny & Rawlings (2002).

Building upon these foundations, this work proposes an integrated framework that addresses the aforementioned challenges to provide a scalable framework for MPC of high-dimensional systems under partial observability. The key enablers of the framework are: (i) OpInf to learn the governing dynamics in a data-driven, non-intrusive manner directly in a latent space spanned by orthogonal POD modes; (ii) a state estimation performed with a UKF, which corrects the open-loop latent-state predictions of the OpInf model using sparse and noisy measurements, and (iii) MPC formulated directly in the latent space, substantially reducing computational cost while retaining nonlinear predictive capability. The framework does not require observing the full state at any moment once the model operators are inferred. This methodology is demonstrated on the one- (Papageorgiou & Smyrlis, 1991) and two-(Kalogirou *et al.*, 2015) dimensional Kuramoto–Sivashinsky equations, canonical benchmarks for chaotic and spatially extended systems. The proposed approach is general and can be extended to multi-dimensional flows or combined with alternative low-dimensional representations, including neural-network-based autoencoders. This is particularly relevant in light of recent applications of latent-space data assimilation using Kalman filters (Amendola *et al.*, 2021; Özalp *et al.*, 2026; Peyron *et al.*, 2021).

A closely related paradigm, particularly given the strong connections between MPC and Reinforcement Learning (RL, Bertsekas, 2024), is the approach proposed by Ozan *et al.* (2025), in which data assimilation with an EnKF and a model learned via Echo State Networks is embedded within an RL control framework. The algorithm proposed in this article can be viewed as an optimisation-based counterpart, where constraints on the control can be explicitly enforced, making it suitable for safety-critical applications.

The remainder of this paper is organised as follows. Section 2 introduces the proposed methodology, detailing the implementation of the MPC formulation, the state estimation procedure based on the UKF, and the data-driven system identification using OpInf. The results obtained for the one-dimensional Kuramoto–Sivashinsky equation are discussed in 3, while the extension to the two-dimensional case is presented in 4. Finally, the main conclusions and perspectives of this work are summarised in 5.

2. Model predictive control from partial measurements

This section introduces the proposed control framework, detailing the modelling strategy based on OpInf, the formulation of MPC, and the application of the UKF for state estimation from sparse and noisy measurements.

The process to be controlled is modelled as an input–output dynamical system characterised by the state vector $\boldsymbol{x}_k \in \mathbb{R}^{n_x}$ sampled at discrete time instants t_k, where n_x denotes the dimension of the state space resulting from a spatial discretisation. The system evolves under the influence of a control input $\boldsymbol{u}_k \in \mathbb{R}^{n_u}$, with n_u representing the dimension of the control space. The discrete-time dynamics are expressed as

$$\boldsymbol{x}_{k+1} = F(\boldsymbol{x}_k, \boldsymbol{u}_k), \tag{1}$$

where $F : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$ represents the nonlinear state-transition function, and $\boldsymbol{x}_0$ denotes the initial state. The discrete-time formulation is obtained via uniform sampling of the continuous-time system with a fixed sampling interval $\Delta t > 0$, such that $t_k = k\Delta t$. Throughout this work, $\boldsymbol{z}_k = \boldsymbol{z}(t_k)$ denotes a generic time-dependent variable $\boldsymbol{z}$ evaluated at t_k.

Formally, the control problem consists of determining the control input $\boldsymbol{u}_k$ at each discrete time instant t_k such that a prescribed performance criterion is optimised subject to system dynamics and problem-dependent constraints. The objective is formulated through a cost function that encodes the desired control goals and which is optimised during the control process. In this work, the focus is on a class of problems where the objective is either to stabilise the system around a fixed reference point or to track a desired trajectory over time. Specifically, at each discrete time instant $t_k,\ k = 1, 2, \ldots$, the controller aims to steer the system state $\boldsymbol{x}_k$ towards a reference $\boldsymbol{x}_k^*$, which may be constant ($i.e.$, $\boldsymbol{x}_k^* = \boldsymbol{x}^*,\ \forall k$) or time-dependent. The cost function penalises deviations from the target state, while possibly also accounting for control effort or other constraints depending on the application.

To address this problem, MPC is employed as a model-based approach. This control technique relies on an explicit representation of the system dynamics to predict its future evolution under a sequence of control actions. These open-loop predictions are used to directly evaluate a cost function, which is then optimised online. The result is an optimal sequence of control actions defined over the prediction horizon. Only the first action in this sequence is applied to the system, while the remaining are discarded. Once the system state is updated, the optimisation is repeated over a shifted horizon. A detailed description of the MPC algorithm, including its formulation and implementation, is presented in Subsection 2.2, while the overall MPC procedure is illustrated schematically on the bottom-left side of Figure 1.

For accurate cost-function evaluation via model rollouts within the MPC loop, it is essential to have a reliable prediction model of the system dynamics. The model must

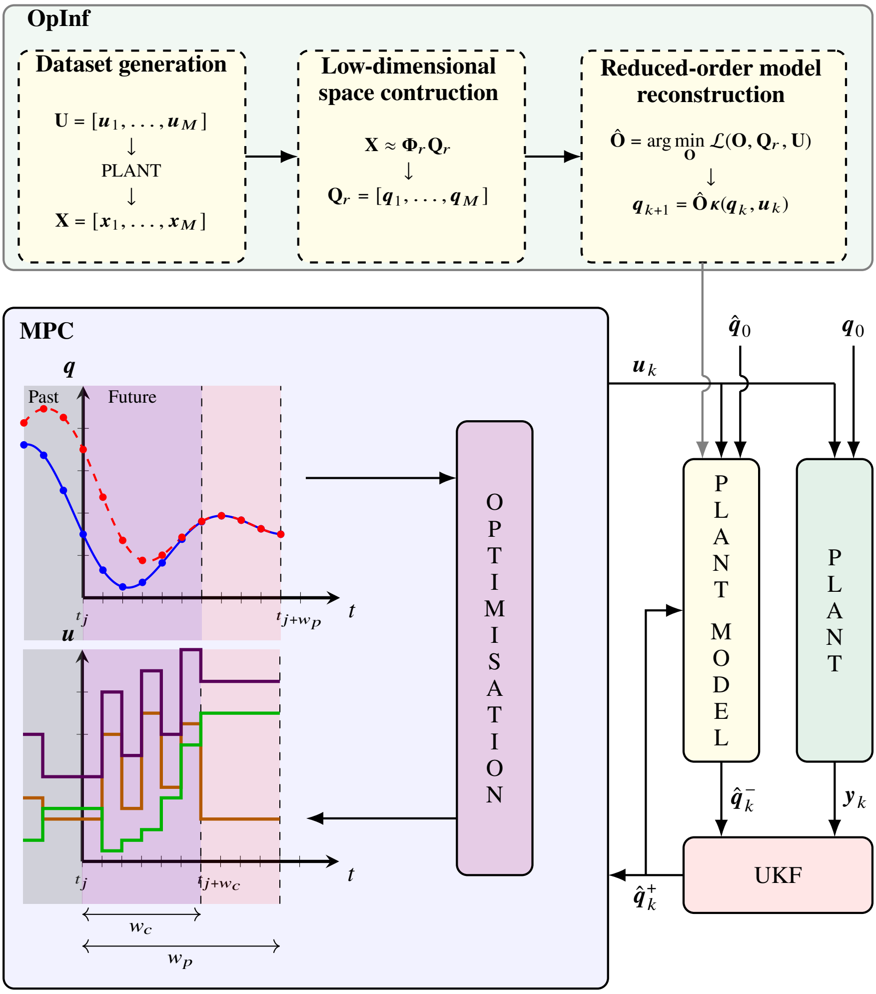

Figure 1: Schematic of the control algorithm. **Top:** Data-driven model identification. An offline dataset of trajectories with M snapshots is collected with open-loop control, and a low-dimensional latent space is constructed from dominant Proper Orthogonal Decomposition modes. The system dynamics are learned directly in this latent space via Operator Inference in a non-intrusive manner and including nonlinear dependencies. **Bottom:** Control procedure. An Unscented Kalman Filter estimates the latent coordinates using partial and noisy state measurements to correct the model predictions. The posterior Kalman estimate is used to initialise the predictive model at each control step. Within Model Predictive Control, the surrogate model generates open-loop latent-state predictions (solid blue line; dots indicate discrete time samples) over a prediction horizon w_p (red shaded area). These predictions are used to optimise the control sequence over the control horizon w_c (purple shaded area). The dashed red line indicates the target latent trajectory.

satisfy two main requirements: (i) it should accurately capture the temporal evolution of the system over the prediction horizon, and (ii) it must be computationally efficient to allow real-time implementation, particularly in high-dimensional settings. In this work, the surrogate model is obtained via OpInf (Kramer *et al.*, 2024). The full-order states are projected onto a low-dimensional subspace defined by the leading POD modes, yielding a latent coordinate $\boldsymbol{q}_k \in \mathbb{R}^r$ at time step t_k, where $r \ll n_x$ denotes the reduced dimension. The system dynamics are then learned directly in the latent space by imposing a polynomial form (see the top of Figure 1). With dynamics defined in the latent space, and owing to the orthonormality of the POD basis and linear mapping, the control problem can be formulated directly in this reduced space, greatly decreasing the computational cost of the MPC implementation.

In practical scenarios, the controller has access to partial and noisy measurements of the state. These are modelled as

$$\boldsymbol{y}_k = C(\boldsymbol{x}_k) + \boldsymbol{\nu}_k^y \quad \in \mathbb{R}^{n_y}, \tag{2}$$

where n_y is the dimensionality of the measurement vector, $C : \mathbb{R}^{n_x} \to \mathbb{R}^{n_y}$ is a known operator that maps the full state to observable quantities, and $\boldsymbol{\nu}_k^y$ denotes the realisation of the measurement noise at time t_k.

To close the feedback loop, a state estimator is then needed to infer the current latent state from the available measurements. In the proposed framework, a UKF is used for this purpose. In particular, this is used in the MPC loop to perform the predictions for the estimation of the cost function. The latent coordinate vector $\boldsymbol{q}_k^-$ coming from the prediction of the Opinf-based model is corrected directly from the sparse and possibly noisy measurements $\boldsymbol{y}_k$ available during control to obtain a posterior Kalman estimate $\boldsymbol{q}_k^+$. This approach accounts for both model uncertainty and measurement noise, enabling accurate prediction of the latent state for MPC rollouts. Unfortunately, unlike the linear case, a separation principle does not generally hold for nonlinear systems, so independent designs of controllers and observers do not guarantee closed-loop stability (Findeisen *et al.*, 2003). However, MPC is inherently robust to small perturbations in the state estimate, and as long as the state estimator performs well, MPC can still be applied effectively in practical scenarios; otherwise, robust MPC implementations (Mayne *et al.*, 2006) are needed.

2.1. *Data-driven plant model estimation*

Among the methodologies useful to derive a surrogate plant model, OpInf provides a non-intrusive data-driven approach. In this framework, the governing equations of the system are neither assumed to be known nor used directly. Instead, the model is inferred purely in a data-driven manner, from snapshots of state trajectories collected from simulations or experiments.

The OpInf procedure can be structured into three main steps: (i) dataset generation, (ii) low-dimensional space construction, and (iii) reduced-order model construction. The resulting ROM can then be integrated within the MPC framework to predict future system states and to optimise control inputs for efficient control.

2.1.1. *Dataset generation*

In this phase, a dataset of states and inputs at different time instants is obtained. Specifically, a set of open-loop control actions is first selected, and the system is then either simulated using the true system's dynamics in (1) or measured experimentally to collect the resulting state trajectories. The dataset, composed of M snapshots collected from one or more open-loop trajectories, is arranged into snapshot matrices as follows:

$$\mathbf{U} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_M] \in \mathbb{R}^{n_u \times M}, \quad \mathbf{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M] \in \mathbb{R}^{n_x \times M} \tag{3}$$

where the matrix $\mathbf{U}$ contains the selected open-loop control inputs and $\mathbf{X}$ the states sampled.

2.1.2. *Low-dimensional space construction*

A reduced-order basis is extracted from the high-dimensional system states using POD. This technique relies on the singular value decomposition of the snapshot matrix $\mathbf{X}$:

$$\mathbf{X} = \mathbf{\Phi}\mathbf{\Sigma}\mathbf{\Psi}^\top, \tag{4}$$

where the superscript $\top$ denotes the transposition operation, $\mathbf{\Phi} = [\phi_1, \ldots, \phi_{n_x}] \in \mathbb{R}^{n_x \times n_x}$ contains the left singular vectors, representing the spatial modes; $\mathbf{\Sigma} = \operatorname{diag}(\sigma_1, \ldots, \sigma_M) \in \mathbb{R}^{n_x \times M}$ is the diagonal matrix of non-negative singular values $\sigma_i,\ i = 1, \ldots, M$, ordered in descending magnitude; and $\mathbf{\Psi} = [\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_M] \in \mathbb{R}^{M \times M}$ contains the right singular vectors, encoding the temporal structure. The columns of $\mathbf{\Phi}$ and $\mathbf{\Psi}$ form orthonormal bases, satisfying $\mathbf{\Phi}^\top\mathbf{\Phi} = \mathbf{I}_{n_x}$ and $\mathbf{\Psi}^\top\mathbf{\Psi} = \mathbf{I}_M$, where $\mathbf{I}_d$ is the identity matrix of dimension d.

By retaining only the first $r \ll \min(n_x, M)$ dominant singular values and vectors, a rank-r approximation of the snapshot matrix is obtained:

$$\mathbf{X} \approx \mathbf{\Phi}_r\mathbf{\Sigma}_r\mathbf{\Psi}_r^\top = \mathbf{\Phi}_r\mathbf{Q}_r, \tag{5}$$

where $\mathbf{\Phi}_r \in \mathbb{R}^{n_x \times r}$, $\mathbf{\Sigma}_r \in \mathbb{R}^{r \times r}$ and $\mathbf{\Psi}_r \in \mathbb{R}^{M \times r}$ are the matrices containing the first r left singular vector, singular values and right singular vectors, respectively. The matrix $\mathbf{Q}_r = \mathbf{\Sigma}_r\mathbf{\Psi}_r^\top \in \mathbb{R}^{r \times M}$ represents the modal amplitudes, obtained by weighting each right singular vector by its corresponding singular value. A common approach for selecting the dimension of the latent space r is to observe the decay of the singular values. Specifically, the retained variance associated with the first r modes is computed as the normalised cumulative sum of the squared singular values $\gamma_r = \sum_{i=1}^{r} \sigma_i^2 / \sum_{i=1}^{\min(n_x,M)} \sigma_i^2$. The value of r is typically chosen such that it exceeds a predefined threshold, ensuring that most of the system's variance is captured (Brindise & Vlachos, 2017; Kramer *et al.*, 2024).

Once r has been selected, the basis matrix $\mathbf{\Phi}_r$, spanning an r-dimensional subspace, defines a low-dimensional coordinate system used to construct the ROM. Each high-dimensional state vector $\boldsymbol{x}_k$ is projected onto the reduced basis as:

$$\boldsymbol{q}_k = \mathbf{\Phi}_r^\top\boldsymbol{x}_k. \tag{6}$$

where $\boldsymbol{q}_k \in \mathbb{R}^r$ corresponds to the low-dimensional representation of $\boldsymbol{x}_k$ and one has used the fact that $\mathbf{Q}_r$ defines an orthonormal basis. A low-order reconstruction of the vector $\boldsymbol{x}_k$, denoted as $\tilde{\boldsymbol{x}}_k$, can then be obtained as

$$\boldsymbol{x}_k \approx \mathbf{\Phi}_r\boldsymbol{q}_k = \tilde{\boldsymbol{x}}_k \in \mathbb{R}^{n_x}. \tag{7}$$

The matrix $\mathbf{Q}_r$ of modal amplitudes is the collection of all the reduced state representations $\mathbf{Q}_r = [\boldsymbol{q}_1, \ldots, \boldsymbol{q}_M]$ in the training dataset.

2.1.3. *Reduced-order model construction*

Once the reduced basis $\boldsymbol{\Phi}_r$ has been identified, the next step consists of learning a dynamical model that evolves the system in the reduced-order space. The discrete-time reduced dynamics to be learned have the following form:

$$\boldsymbol{q}_{k+1} = F_q(\boldsymbol{q}_k, \boldsymbol{u}_k), \tag{8}$$

where $F_q : \mathbb{R}^r \times \mathbb{R}^{n_u} \to \mathbb{R}^r$ describes the temporal evolution of the latent coordinates and must be inferred from data. Operator Inference provides a data-driven framework to identify F_q by fitting a polynomial model to snapshot data. In particular, the inference is performed in a polynomial feature space:

$$\boldsymbol{q}_{k+1} = \boldsymbol{c} + \mathbf{A}\boldsymbol{q}_k + \mathbf{H}(\boldsymbol{q}_k \circledast \boldsymbol{q}_k) + \mathbf{G}(\boldsymbol{q}_k \circledast \boldsymbol{q}_k \circledast \boldsymbol{q}_k) + \mathbf{B}\boldsymbol{u}_k + \mathbf{N}(\boldsymbol{q}_k \circledast \boldsymbol{u}_k) + \ldots, \tag{9}$$

where $\boldsymbol{c}$ is a constant offset, $\mathbf{A}$ is the linear operator, $\mathbf{H}$ and $\mathbf{G}$ encode quadratic and cubic nonlinearities, respectively, $\mathbf{B}$ and $\mathbf{N}$ model the input and state-input interaction terms, and other terms can be included. The symbol $\circledast$ denotes the symmetrised Kronecker product used to construct higher-order tensor features. For $\boldsymbol{q} \in \mathbb{R}^r$, the degree-$n$ polynomial term, $\boldsymbol{q} \circledast \overset{n}{\cdots} \circledast \boldsymbol{q}$, is a vector of dimension $\binom{r+n-1}{n}$.

To express this model compactly, a feature operator $\boldsymbol{\rho} : \mathbb{R}^r \times \mathbb{R}^{n_u} \to \mathbb{R}^m$ is defined, where m represents the total number of polynomial features constructed from the latent variables and control inputs up to the chosen polynomial order. This operator maps the latent state and control input to the corresponding polynomial basis:

$$\boldsymbol{\rho}(\boldsymbol{q}_k, \boldsymbol{u}_k) := \begin{bmatrix} 1 \\ \boldsymbol{q}_k \\ \boldsymbol{q}_k \circledast \boldsymbol{q}_k \\ \boldsymbol{q}_k \circledast \boldsymbol{q}_k \circledast \boldsymbol{q}_k \\ \boldsymbol{u}_k \\ \boldsymbol{q}_k \circledast \boldsymbol{u}_k \\ \vdots \end{bmatrix} \in \mathbb{R}^m. \tag{10}$$

The reduced-order dynamics can then be written as

$$\boldsymbol{q}_{k+1} = \mathbf{O}\,\boldsymbol{\rho}(\boldsymbol{q}_k, \boldsymbol{u}_k), \tag{11}$$

where $\mathbf{O} \in \mathbb{R}^{r\times m}$ is the matrix, to be identified, that collects all the operators from (9) and is structured as

$$\mathbf{O} = \begin{bmatrix} \boldsymbol{c} & \mathbf{A} & \mathbf{H} & \mathbf{G} & \mathbf{B} & \mathbf{N} & \ldots \end{bmatrix}. \tag{12}$$

To infer $\mathbf{O}$ from data, the feature operator $\boldsymbol{\rho}$ is evaluated offline on snapshots $(\boldsymbol{q}_k, \boldsymbol{u}_k)$ to form the design matrix:

$$\mathbf{D} = \begin{bmatrix} \boldsymbol{\rho}(\boldsymbol{q}_1, \boldsymbol{u}_1)^\top \\ \vdots \\ \boldsymbol{\rho}(\boldsymbol{q}_{M-1}, \boldsymbol{u}_{M-1})^\top \end{bmatrix} \in \mathbb{R}^{(M-1)\times m}, \tag{13}$$

and the corresponding target matrix:

$$\mathbf{Z} = \begin{bmatrix} \boldsymbol{q}_2^\top \\ \vdots \\ \boldsymbol{q}_M^\top \end{bmatrix} \in \mathbb{R}^{(M-1)\times r}. \tag{14}$$

The OpInf problem is then described by the following regularised least-squares optimisation:

$$\hat{\mathbf{O}} = \arg\min_{\mathbf{O}} \|\mathbf{D}\mathbf{O}^\top - \mathbf{Z}\|_F^2 + \lambda_o \|\mathbf{O}\|_F^2, \tag{15}$$

where $\lambda_o \geq 0$ is a regularization parameter and $\|\cdot\|_F$ denotes the Frobenius norm. The first term in (15) promotes model accuracy with respect to the training data, while the second discourages overfitting by penalising large operator coefficients. In particular, the regularisation term acts as a bias in the optimisation problem that guides learning when data are scarce or noisy. This promotes meaningful models with reasonable predictive capabilities that can generalise well to unseen inputs and initial conditions.

Once the optimal operator $\hat{\mathbf{O}}$ has been identified, the OpInf model is defined by:

$$\hat{\boldsymbol{q}}_{k+1} = \hat{\mathbf{O}}\, \boldsymbol{\rho}(\hat{\boldsymbol{q}}_k, \boldsymbol{u}_k), \tag{16}$$

where $\hat{\boldsymbol{q}}_k$ denotes the estimated latent states from the reduced-order model, and the corresponding full-state reconstruction is obtained as $\hat{\boldsymbol{x}}_k = \boldsymbol{\Phi}_r \hat{\boldsymbol{q}}_k$.

2.2. *Model Predictive Control*

In this work, MPC is formulated directly in terms of the latent variables identified via OpInf. The following formulation assumes access to estimated latent states at each control step and leverages the reduced-order dynamics for control purposes.

In particular, starting from time instants t_k, given an estimate of the current latent coordinate $\hat{\boldsymbol{q}}_k$, the model predicts future latent coordinates $\hat{\boldsymbol{q}}_{k+j|k}$ under a sequence of control actions over a prediction horizon $t_{k+j}, j = 1, \ldots, w_p$. The notation $\hat{\boldsymbol{q}}_{k+j|k}$ indicates a prediction of the latent coordinate at time step t_{k+j} obtained advancing the learned predictive model in (16) from the latent state estimate $\hat{\boldsymbol{q}}_k$, such that $\hat{\boldsymbol{q}}_{k|k} := \hat{\boldsymbol{q}}_k$.

The optimal input sequence $\{\boldsymbol{u}_{k+j}^{\text{opt}}\}_{j=0}^{w_c-1}$ is then obtained by minimising in a control window $t_{k+j}, j = 0, \ldots, w_c - 1$, the cost function

$$\begin{aligned} J(\hat{\boldsymbol{u}}_k, \ldots, \hat{\boldsymbol{u}}_{k+w_c-1}, \hat{\boldsymbol{q}}_k) = &\sum_{j=0}^{w_p-1} \|\hat{\boldsymbol{q}}_{k+j|k} - \boldsymbol{q}^*_{k+j}\|^2_{\mathbf{R}_q} + V_f(\hat{\boldsymbol{q}}_{k+w_p|k}) \\ &+ \sum_{j=0}^{w_c-1} \left(\|\hat{\boldsymbol{u}}_{k+j}\|^2_{\mathbf{R}_u} + \|\Delta\hat{\boldsymbol{u}}_{k+j}\|^2_{\mathbf{R}_{\Delta u}}\right), \end{aligned} \tag{17}$$

with respect to the actions $\hat{\boldsymbol{u}}_k, \ldots, \hat{\boldsymbol{u}}_{k+w_c-1}$ used to build it via rollout. Here, $\|\boldsymbol{d}\|^2_{\mathbf{K}} = \boldsymbol{d}^\top \mathbf{K} \boldsymbol{d}$ is the weighted norm of a generic vector $\boldsymbol{d}$ with respect to a symmetric and positive definite matrix $\mathbf{K}$. The matrices $\mathbf{R}_q$, $\mathbf{R}_u$, and $\mathbf{R}_{\Delta u}$ are weighting matrices assigning relative importance to each term in the cost function, with $\mathbf{R}_q$ typically positive semidefinite and $\mathbf{R}_u$, $\mathbf{R}_{\Delta u}$ positive definite. The function V_f represents a terminal cost accounting for the desired performance at the end of the prediction horizon. This term captures contributions to the overall cost not explicitly considered in the accumulated running cost and renders MPC a local, model-based approximation of the optimal Bellman policy

(Bertsekas, 2024). The term $\Delta\hat{\boldsymbol{u}}_{k+j|k} = \hat{\boldsymbol{u}}_{k+j|k} - \hat{\boldsymbol{u}}_{k+j-1|k}$ is the input variability in time. In the cost function, $\boldsymbol{q}_k^*$ denotes the low dimensional representation of the target state $\boldsymbol{x}^*$ at time t_k, obtained via projection $\boldsymbol{q}_k^* = \boldsymbol{\Phi}_r^\top \boldsymbol{x}_k^*$, as described in (6). In general, $w_c \leq w_p$; when $w_c < w_p$, the control input is maintained constant beyond the control horizon.

After the optimisation, only the first control input in the sequence, $\boldsymbol{u}_k = \boldsymbol{u}_k^{\mathrm{opt}}$, is applied to the system, and the process identically repeats at the next time step and with the new latent state estimate available. MPC allows a straightforward introduction of state or input constraints within the optimisation in (17). For instance, indicating with superscript (i) the ith component of a vector, constraints on the input are considered:

$$\hat{\boldsymbol{u}}_k^{(i)} \in [\boldsymbol{u}_{\min}, \boldsymbol{u}_{\max}], \quad \Delta\hat{\boldsymbol{u}}_k^{(i)} \in [\Delta\boldsymbol{u}_{\min}, \Delta\boldsymbol{u}_{\max}] \quad \forall\, i, k. \tag{18}$$

The MPC loop is then summarised in Alg. Algorithm 2.

Algorithm 2 MPC Optimization Problem

1: Collect the latent state estimate $\hat{\boldsymbol{q}}_k$ at time step t_k

2: Solve the optimisation:

$$\boldsymbol{u}_k^{\mathrm{opt}}, \ldots, \boldsymbol{u}_{k+w_c-1}^{\mathrm{opt}} = \min_{\{\hat{\boldsymbol{u}}_k, \ldots, \hat{\boldsymbol{u}}_{k+w_c-1}\}} J(\hat{\boldsymbol{u}}_k, \ldots, \hat{\boldsymbol{u}}_{k+w_c-1}, \hat{\boldsymbol{q}}_k)$$

Subject to constraints:

$$\begin{array}{ll}
\hat{\boldsymbol{u}}_{k+j}^{(i)} \in [\boldsymbol{u}_{\min}^{(i)}, \boldsymbol{u}_{\max}^{(i)}], & j = 0, \ldots, w_c - 1, \quad i = 1, \ldots, m \\
\Delta\hat{\boldsymbol{u}}_{k+j}^{(i)} \in [\Delta\boldsymbol{u}_{\min}^{(i)}, \Delta\boldsymbol{u}_{\max}^{(i)}], & j = 0, \ldots, w_c - 1, \quad i = 1, \ldots, m \\
\hat{\boldsymbol{u}}_{k+j} = \hat{\boldsymbol{u}}_{k+w_c-1}, & j \geq w_c \\
\hat{\boldsymbol{q}}_{k|k} := \hat{\boldsymbol{q}}_k, & \\
\hat{\boldsymbol{q}}_{k+j+1|k} = \hat{F}_q(\hat{\boldsymbol{q}}_{k+j|k}, \hat{\boldsymbol{u}}_{k+j}), & j = 0, \ldots, w_p - 1
\end{array}$$

3: Apply $\boldsymbol{u}_k = \boldsymbol{u}_k^{\mathrm{opt}}$

4: Update the time step to t_{k+1} and repeat from step 1

For all cases considered in this work, the optimisation problem in (17) is solved using CasADi (Andersson *et al.*, 2019), a symbolic framework for algorithmic differentiation that enables the efficient implementation of numerical optimal control methods, including nonlinear MPC. In particular, the optimisation problems are solved using IPOPT (Interior Point OPTimizer, Wächter & Biegler, 2006), a large-scale nonlinear solver based on a primal-dual interior-point method.

2.3. *Data assimilation with UKF*

Once the plant model has been identified and the MPC framework established, it is necessary to initialise the predictive model with an appropriate estimate of the system's latent state $\hat{\boldsymbol{q}}_k$, at each control time step. This is essential to evaluate the cost function in (17) via plant model rollouts. In realistic scenarios, MPC typically operates under partial observability, where only a measure $\boldsymbol{y}_k$ of the system state is available at time t_k. Consequently, reconstructing the full latent state vector $\boldsymbol{q}_k$ from $\boldsymbol{y}_k$ represents a state estimation problem. Note that measurements need not be collected at every control step; the estimation can propagate the state in between the updates using the model dynamics. To deal with this estimation problem, an Unscented Kalman Filter (Julier & Uhlmann, 1997; Wan & Van Der Merwe, 2000) is employed. The UKF serves as a data assimilation layer, providing an online estimation of the latent state by combining the predictive reduced order model from Opinf in (16) with the available observations. At each control time, the UKF first propagates the prior estimate of the latent coordinate forward in time using the model dynamics and then corrects this prediction using the current observation $\boldsymbol{y}_k$. This process allows a robust initialisation of the MPC at each decision step.

Although the true underlying system is deterministic, the learned dynamical model is an approximation subject to modelling errors, due to limited training data, truncation of the reduced-order basis, unmodeled dynamics, or imperfect identification of nonlinear interactions. As a result, these prediction errors can be represented as a stochastic process noise term, leading to the modified latent dynamics

$$\hat{\boldsymbol{q}}_{k+1} = \hat{F}_q(\hat{\boldsymbol{q}}_k, \boldsymbol{u}_k) + \boldsymbol{\nu}_k^q \tag{19}$$

where $\boldsymbol{\nu}_k^q$ is a realisation of the process noise at time t_k, representing the stochastic component of the model and accounting for unmodeled residuals.

The Kalman Filter addresses the state estimation problem by assuming the latent coordinate is a Gaussian random variable. The corrected estimate at time t_k is computed as

$$\hat{\boldsymbol{q}}_k^+ = \hat{\boldsymbol{q}}_k^- + \boldsymbol{K}_k \left(\boldsymbol{y}_k - \hat{\boldsymbol{y}}_k\right), \tag{20}$$

where $\hat{\boldsymbol{q}}_k^-$ is the prior latent state prediction obtained by integrating the OpInf surrogate model in (16) and $\hat{\boldsymbol{y}}_k$ is the predicted current observation derived through the measurement function defined in (2). The matrix $\boldsymbol{K}_k \in \mathbb{R}^{r \times n_y}$ is the Kalman gain and determines how much the prior estimate is adjusted in response to the observed innovation $(\boldsymbol{y}_k - \hat{\boldsymbol{y}}_k)$.

Following the Kalman Filter theory (see Welch & Bishop (1995)), the optimal computation of the Kalman gain is given by:

$$\boldsymbol{K}_k = \mathbf{P}_{qy}\mathbf{P}_{yy}^{-1}, \tag{21}$$

where $\mathbf{P}_{qy} \in \mathbb{R}^{r \times n_y}$ and $\mathbf{P}_{yy} \in \mathbb{R}^{n_y \times n_y}$ denote the cross-covariance matrices between the latent state and predicted measurement, and the covariance of the predicted measurement, respectively. Here, $\mathbf{P}_{yy}^{-1}$ denotes the inverse of the matrix $\mathbf{P}_{yy}$. In linear systems, these quantities can be computed analytically, yielding an exact recursive solution. However, for nonlinear systems, these covariances involve expectations over nonlinear transformations of random variables and cannot be obtained in closed form. The Extended Kalman Filter (EKF, Ribeiro, 2004) addresses this by linearising the dynamics and observation models about the current estimate, using a first-order Taylor expansion to approximate the required covariances. This approach may lead to sub-optimal performance or even divergence of the filter, since the first-order linearization does not accurately capture the true nonlinear transformations of the mean and covariance, resulting in incorrect propagation of the state uncertainty (Marafioti *et al.*, 2009).

To overcome these limitations, the UKF uses the unscented transformation (UT), which deterministically selects a set of points (called sigma points) that are propagated through the nonlinear model and measurement functions. These sample points completely capture the true mean and covariance of the latent state and, when propagated through the nonlinear system, capture the posterior mean and covariance accurately to the third order for any nonlinearity. Alternatively, the EnKF (EnKF, Evensen, 2003) employs a Monte Carlo approach that propagates an ensemble of system realisations to estimate the posterior statistics. The EnKF has been used in a similar control problem in reinforcement learning (Ozan *et al.*, 2025).

The UKF estimation process comprises four main steps: (i) filter initialisation, (ii) sigma point generation, (iii) time update, and (iv) measurement update and correction.

2.3.1. *Filter initialisation*

To incorporate both process and measurement noise within the estimation framework, an augmented state vector is defined at each time step t_k:

$$\boldsymbol{q}_k^a = \begin{bmatrix} \boldsymbol{q}_k \\ \boldsymbol{\nu}_k^q \\ \boldsymbol{\nu}_k^y \end{bmatrix} \in \mathbb{R}^{n_a}, \tag{22}$$

with $n_a = 2r + n_y$ the augmented dimension. Similarly, the augmented state covariance matrix is built as

$$\mathbf{P}_k^a = \begin{bmatrix} \mathbf{P}_k^+ & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{R} \end{bmatrix} \in \mathbb{R}^{n_a \times n_a}, \tag{23}$$

where $\mathbf{P}_k^+ \in \mathbb{R}^{r\times r}$ represents the posterior covariance matrix of the latent state, while $\mathbf{Q} \in \mathbb{R}^{r\times r}$ and $\mathbf{R} \in \mathbb{R}^{n_y \times n_y}$ denote the known covariance matrices of the process and measurement noise, respectively.

At the initial time t_0, the UKF is started with an initial estimate of the latent coordinate posterior mean, $\hat{\boldsymbol{q}}_0^+$ and covariance matrix $\mathbf{P}_0^+$:

$$\hat{\boldsymbol{q}}_0^+ = \mathbb{E}[\boldsymbol{q}_0] \tag{24}$$

$$\mathbf{P}_0^+ = \mathbb{E}[(\boldsymbol{q}_0 - \hat{\boldsymbol{q}}_0^+)(\boldsymbol{q}_0 - \hat{\boldsymbol{q}}_0^+)^\top], \tag{25}$$

where $\mathbb{E}[\cdot]$ denotes the expectation operator. Since these latent quantities are not directly observable, they are obtained indirectly. First, the full initial state $\hat{\boldsymbol{x}}_0^+$ is estimated from $\boldsymbol{y}_0$ using Gaussian Process Regression (GPR, Schulz *et al.*, 2018). This estimate is then projected onto the reduced basis to yield the latent initial state $\hat{\boldsymbol{q}}_0^+ = \boldsymbol{\Phi}_r^\top \hat{\boldsymbol{x}}_0^+$. The corresponding latent covariance, $\mathbf{P}_0^+$, is derived from the uncertainty provided by the GPR model, ensuring a consistent initialisation for recursive filtering.

Then, the augmented state is initialised as

$$\hat{\boldsymbol{q}}_0^a = [(\hat{\boldsymbol{q}}_0^+)^\top \quad \mathbf{0}_r^\top \quad \mathbf{0}_{n_y}^\top]^\top, \tag{26}$$

where $\mathbf{0}_d$ denotes a zeros vector of dimension d. Similarly, the augmented state covariance matrix is

$$\mathbf{P}_0^a = \begin{bmatrix} \mathbf{P}_0^+ & 0 & 0 \\ 0 & \mathbf{Q} & 0 \\ 0 & 0 & \mathbf{R} \end{bmatrix}. \tag{27}$$

2.3.2. *Sigma points generation*

A set of $2n_a+1$ sigma points is generated to approximate the distribution of the augmented state:

$$\begin{aligned} \boldsymbol{\mathcal{Q}}_{k-1}^{a,(0)} &= \hat{\boldsymbol{q}}_{k-1}^a, \\ \boldsymbol{\mathcal{Q}}_{k-1}^{a,(i)} &= \hat{\boldsymbol{q}}_{k-1}^a + \sqrt{n_a + \lambda}\, \mathbf{S}_{(\cdot,i)}, \quad \forall\, i = 1, \dots, n_a, \\ \boldsymbol{\mathcal{Q}}_{k-1}^{a,(i)} &= \hat{\boldsymbol{q}}_{k-1}^a - \sqrt{n_a + \lambda}\, \mathbf{S}_{(\cdot,i)}, \quad \forall\, i = n_a+1, \dots, 2n_a, \end{aligned} \tag{28}$$

where $\mathbf{S}$ is the Cholesky factor of $\mathbf{P}^a_{k-1}$ and $\lambda = \alpha^2(n_a + \kappa) - n_a$, with α generally a small value $0 < \alpha \leq 1$ and $\kappa \geq 0$ tuning parameters controlling the spread of the sigma points.

For simplicity, the sigma points are partitioned into these components:

$$\boldsymbol{\mathcal{Q}}^{a,(i)}_{k-1} = \begin{bmatrix} \boldsymbol{\mathcal{Q}}^{q,(i)}_{k-1} \\ \boldsymbol{\mathcal{Q}}^{\nu^q,(i)}_{k-1} \\ \boldsymbol{\mathcal{Q}}^{\nu^y,(i)}_{k-1} \end{bmatrix}. \tag{29}$$

where the superscripts q, ν^q, ν^y refer to a partition conformal to the dimensions of the latent state, process noise, and measurement noise, respectively.

2.3.3. *Time update*

Each sigma point is propagated through the dynamic model:

$$\boldsymbol{\mathcal{Q}}^{q-,(i)}_{k} = \hat{F}_q(\boldsymbol{\mathcal{Q}}^{q,(i)}_{k-1}, \boldsymbol{u}_{k-1}) + \boldsymbol{\mathcal{Q}}^{\nu^q,(i)}_{k-1}, \quad \forall i = 0, \ldots, 2n_a + 1, \tag{30}$$

where $\boldsymbol{\mathcal{Q}}^{q-,(i)}_{k}$ denotes the one-step prediction of the ith sigma point $\boldsymbol{\mathcal{Q}}^{a,(i)}_{k-1}$ using the surrogate model of the dynamics in (16). The predicted prior mean $\hat{\boldsymbol{q}}^-_k$ and covariance $\mathbf{P}^-_k$ of the latent state at time k are approximated by weighted sums of the propagated sigma points:

$$\hat{\boldsymbol{q}}^-_k = \sum_{i=0}^{2n_a} W^m_i \boldsymbol{\mathcal{Q}}^{q-,(i)}_{k} \tag{31}$$

$$\mathbf{P}^-_k = \sum_{i=0}^{2n_a} W^c_i \left[\boldsymbol{\mathcal{Q}}^{q-,(i)}_{k} - \hat{\boldsymbol{q}}^-_k\right] \left[\boldsymbol{\mathcal{Q}}^{q-,(i)}_{k} - \hat{\boldsymbol{q}}^-_k\right]^\top \tag{32}$$

where the weighting coefficients are defined as

$$W^m_0 = \frac{\lambda}{n_a + \lambda} \tag{33}$$

$$W^c_0 = \frac{\lambda}{n_a + \lambda} + (1 - \alpha^2 + \beta) \tag{34}$$

$$W^m_i = W^c_i = \frac{1}{2(n_a + \lambda)} \quad \forall i = 1, \ldots, 2n_a \tag{35}$$

being β a parameter incorporating prior knowledge of the distribution of the state. For Gaussian distributions, $\beta = 2$ is optimal.

2.3.4. *Measurement update and correction*

Each sigma point is transformed through the measurement function included in (2) to the observation space:

$$\boldsymbol{y}^{-,(i)}_{k} = C(\boldsymbol{\Phi}_r \boldsymbol{\mathcal{Q}}^{q-,(i)}_{k}) + \boldsymbol{\mathcal{Q}}^{\nu^y,(i)}_{k-1} \tag{36}$$

The predicted measurement mean, covariance and cross-covariance are then computed as:

$$\hat{\boldsymbol{y}}_k^- = \sum_{i=0}^{2n_a} W_i^m \boldsymbol{y}_k^{-,(i)} \tag{37}$$

$$\mathbf{P}_{yy,k} = \sum_{i=0}^{2n_a} W_i^c [\boldsymbol{y}_k^{-,(i)} - \hat{\boldsymbol{y}}_k^-][\boldsymbol{y}_k^{-,(i)} - \hat{\boldsymbol{y}}_k^-]^\top \tag{38}$$

$$\mathbf{P}_{qy,k} = \sum_{i=0}^{2n_a} W_i^c [\boldsymbol{\mathcal{Q}}_k^{q-,(i)} - \hat{\boldsymbol{q}}_k^-][\boldsymbol{y}_k^{-,(i)} - \hat{\boldsymbol{y}}_k]^\top \tag{39}$$

Finally, the Kalman gain matrix is computed as

$$\mathbf{K}_k = \mathbf{P}_{qy,k}\mathbf{P}_{yy,k}^{-1}, \tag{40}$$

which is used to update the posterior estimate of the latent state mean and covariance conditioned on the actual measurement $\boldsymbol{y}_k$:

$$\hat{\boldsymbol{q}}_k^+ = \hat{\boldsymbol{q}}_k^- + \mathbf{K}_k \left(\boldsymbol{y}_k - \hat{\boldsymbol{y}}_k^-\right), \tag{41}$$

$$\mathbf{P}_k^+ = \mathbf{P}_k^- - \mathbf{K}_k \mathbf{P}_{yy,k} \mathbf{K}_k^\top. \tag{42}$$

This procedure is applied recursively for each control step, with measurement corrections performed only when new observations are available.

3. Control of the 1D Kuramoto-Sivashinsky

The proposed control framework is now demonstrated on the one-dimensional Kuramoto-Sivashinsky (1D KS) equation, a nonlinear partial differential equation that exhibits spatiotemporal chaos. It was first introduced as a model for laminar flame front instabilities (Sivashinsky, 1980; Siv Ashinsky, 1988) and chemical reaction-diffusion processes (Kuramoto, 1978). The equation is generally used as a prototypical model for chaotic dynamics, as it shares certain features characteristic of the Navier–Stokes equations (Dankowicz *et al.*, 1996).

Let $x(\xi, t)$ be the system state, where ξ is the spatial coordinate. The temporal evolution of the system can be described by the following equation:

$$\frac{\partial x}{\partial t} + x\frac{\partial x}{\partial \xi} + \frac{\partial^2 x}{\partial \xi^2} + \frac{\partial^4 x}{\partial \xi^4} = 0 \tag{43}$$

The equation includes a nonlinear convection term, a diffusion term, and a hyperdiffusion term, respectively. The 1D KS is defined on the spatial domain $\xi \in [0, L]$, with $L = 22$, and periodic boundary conditions $x(\xi, t) = x(\xi + L, t)$. The choice of $L = 22$, following Bucci *et al.* (2019), ensures that the 1D KS dynamics exhibit features similar to those observed in the Navier-Stokes equations. To enable control, a spatiotemporal forcing term $\Gamma(\xi, t)$ is added in the right-hand side of the equation, acting as the control input (Bucci *et al.*, 2019):

$$\Gamma(\xi, t) = \sum_{i=1}^{4} u^{(i)}(t) \frac{1}{\sqrt{2\pi\sigma_a^2}} \exp\left(-\frac{(\xi - \xi_a^{(i)})^2}{2\sigma_a^2}\right), \tag{44}$$

where $\xi_a^{(i)} = \frac{iL}{4}$ denote the actuators locations, $\sigma_a = 0.4$ defines the width of the Gaussian support, and $u^{(i)}(t)$ is the time-dependent amplitude of the i-th actuator. The sensors are assumed to be equispaced along the spatial domain as follows:

$$\xi_s^{(i)} = 1 + \frac{iL}{n_y}, \quad i = 0, \dots, n_y - 1. \tag{45}$$

The controlled 1D KS equation is integrated using a Fourier expansion of the states, a procedure enabled by the periodic boundary conditions. For this purpose, $n_x = 64$ collocation points and an integration time step of $\Delta t = 0.05$ are used. This implementation follows the approach described in Bucci *et al.* (2019), to which the reader is referred for further details.

The control objective is to stabilise the system state around its nontrivial invariant solutions, denoted as E_1, E_2, and E_3 (Papageorgiou & Smyrlis, 1991). These equilibria have previously been employed as control targets in reinforcement learning applications, as reported in Bucci *et al.* (2019).

3.1. *Reduced-order surrogate model*

The first step in constructing a reduced-order model for the 1D KS equation is to generate a representative training dataset under both uncontrolled and controlled conditions. Specifically, a single trajectory of the full 1D KS equation is simulated over a total time of $T_{\mathrm{tr}} = 1000$ time units (t.u.), with the first 200 t.u. corresponding to the uncontrolled system and the remaining portion subject to an open-loop control sequence. For both dataset generation and subsequent model identification and control, a time step that is an integer multiple of the numerical integration step of the 1D KS solver is used, resulting in $\Delta t_c = 0.1$ t.u. Snapshots are therefore collected every Δt_c, yielding a total of $M = 10{,}000$ snapshots in the training set. In the controlled part of the training dataset, the forcing signal for each actuator is generated as a realisation of a Gaussian white-noise process, where independent samples are drawn from a normal distribution at each time step. The time series is then smoothed in the frequency domain using a Fourier filter with cut-off $f_{co} = 1$ to eliminate high-frequency components and avoid abrupt actuation changes. Finally, each filtered signal is normalised to have zero mean and a standard deviation of $\sigma_u = 3$. A visualisation of the training dataset, including both system states and corresponding control actions, is presented in the left column of Figure 2.

The training dataset is used to identify the predictive dynamical model using OpInf. First, the POD is applied to extract a low-dimensional basis. The truncation rank r is selected such that the cumulative variance is $\gamma_r = 0.9999$, resulting in a reduced-order subspace of dimension $r = 14$. The reduced-order model is then identified by solving the optimisation problem in (15). A polynomial structure is imposed on the latent dynamics, including in the operator matrix in (12) terms up to third order in the state variables. Specifically, the model includes: a constant term $\mathbf{c}$, a linear operator $\mathbf{A}$, a quadratic operator $\mathbf{H}$, and a cubic operator $\mathbf{G}$. The dependence on the control input is modelled linearly through the operator $\mathbf{B}$, and the second-order state-input interaction is captured by the operator $\mathbf{N}$. The optimisation is performed with a regularisation coefficient $\lambda_o = 0.886$, determined via grid search by minimising the prediction error over the training dataset as done in Kramer *et al.* (2024). The identified model exhibits a relative one-step prediction bias of 0.07 due to truncation in the POD modes during system identification. This bias was computed as the mean one-step prediction error over an ensemble of initial conditions, divided by the root-mean-square error (RMSE) and averaged across all state components. Although this bias is small and can be considered

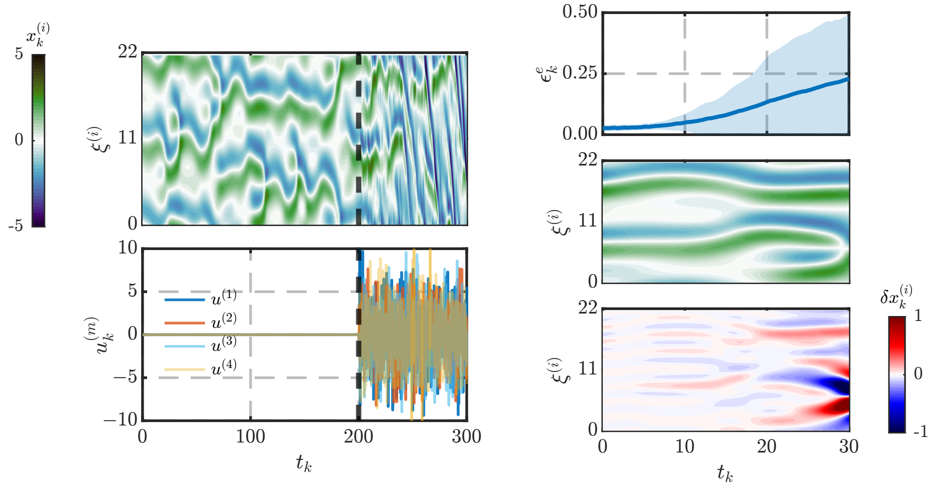

Figure 2: Left panels show the training dataset used for system identification via operator inference. The top row displays the state map, *i.e.* the spatio-temporal evolution of the state, while the bottom row shows the open-loop actuation. The training dataset is shown for the first 300 *t.u.*. The right panels show the model evaluation on an ensemble of 250 trajectories. The top row presents the prediction error (ϵ_k^e) over time, with the average as a solid line and the $\pm\sigma$ confidence region shaded. The middle and bottom rows show the predicted state and the prediction error for a trajectory with median ensemble error. Each prediction starts from the ground-truth state vector.

negligible for the current model, cases with larger systematic errors would require the use of bias-aware data assimilation strategies (Dee, 2005; Nóvoa *et al.*, 2024).

After the surrogate model is identified, its generalisability is tested on a validation dataset with the same structure as the training dataset. The model is evaluated for an ensemble of $N = 250$ trajectories with different initial conditions and open-loop actuations. The prediction performance is quantified using a normalised RMSE:

$$\epsilon_k^e = \frac{\sqrt{\frac{1}{n_x}\sum_{i=1}^{n_x}\left(\hat{x}_k^{(i)} - x_k^{(i)}\right)^2}}{\sqrt{\frac{1}{n_x}\sum_{i=1}^{n_x}\left(x_k^{(i)}\right)^2}}, \tag{46}$$

where $x_k^{(i)}$ and $\hat{x}_k^{(i)}$ denote the ith component of the true and predicted state vectors at time step t_k. For this case, the true initial condition to perform the open-loop prediction is assumed to be known. As shown in Figure 2, the model provides accurate predictions even for mid- to long-term horizons, with the ensemble mean error remaining below 10% for prediction windows of up to 10 *t.u.*. It should be noted that at the initial time instant, the errors are nonzero due to the low-order reconstruction losses obtained by projecting the true initial condition onto the latent space and subsequently reprojecting it into the physical space, i.e., $\hat{\boldsymbol{x}}_0 = \boldsymbol{\Phi}_r\boldsymbol{\Phi}_r^\top\boldsymbol{x}_0$. The contained ensemble sampled standard deviation of the error also suggests that the model systematically produces accurate predictions for horizons smaller than 10 $t.u.$. The prediction capabilities of the model are also proved by the median error case prediction in Figure 2, where the prediction error map ($\delta x_k^{(i)} = \hat{x}_k^{(i)} - x_k^{(i)}$) is shown. Prediction errors start to accumulate for horizons longer than 20 t.u., due to neglected terms in the dynamics and unmodeled effects arising from the truncation to a low-dimensional subspace. This phenomenon is particularly pronounced in chaotic systems such as the 1D KS equation, where long-term prediction is inherently challenging.

3.2. *State estimation via UKF*

The UKF is now employed to estimate both the latent and full system states by combining the surrogate model predictions with available measurement data. Figure 3 illustrates the application of the UKF for the state estimation problem. The full state is advanced by the OpInf model and corrected by measurements from equispaced sensors located in the domain, as given by (45). The sensors measure the state vector at the given location with a certain noise level, as defined in (2), and available at constant time intervals, denoted as T_s. The measurement noise is assumed to be white Gaussian, $\nu^y \sim \mathcal{N}(0, \sigma_{\nu^y})$, with standard deviation σ_{ν^y}. The left column of the figure shows the case with $n_y = 4$ equispaced sensors, $\sigma_{\nu^y} = 0.1$, and measures available at every estimation time steps, leading to $T_s = \Delta t_c = 0.1$. The actuation considered during this simulation follows the same filtered Gaussian random pattern as in the training dataset, but constitutes a separate validation dataset.

For the UKF, prior estimates of the process and measurement noise covariances, $\mathbf{Q}$ and $\mathbf{R}$, must be specified. These matrices determine the relative weighting between model predictions and measurement corrections: a larger $\mathbf{Q}$ reduces reliance on the model, while a larger $\mathbf{R}$ reduces trust in the measurements. In this work, the process noise covariance is chosen as a diagonal matrix, $\mathbf{Q} = 0.007\mathbf{I}_{n_q}$, and the measurement noise covariance is set as $\mathbf{R} = \sigma_{\nu^y}^2 \mathbf{I}_{n_q}$, with the measurement variance assumed known. Both covariance matrices are manually set and kept constant during the filter execution. However, automatic procedures for selecting the covariance matrices and the Unscented Transform hyperparameters have been proposed in the literature (Scardua & da Cruz, 2017), together with adaptive UKF formulations in which $\mathbf{Q}$ and $\mathbf{R}$ are updated online to account for variations in the noise statistics (Zhang *et al.*, 2022; Jiang *et al.*, 2007), but these methods are not further explored here. The unscented transformation in UKF is also configured with the parameters $\alpha = 0.1$, $\beta = 2.0$, and $\kappa = 0$.

In the estimation process, the initial condition is assumed unknown. The model for data assimilation with UKF is initialised using an initial guess obtained via GPR based on the available partial measurements at the initial time step. By having the initial measurement $\boldsymbol{y}_0$, the GPR provides an initial state estimate $\boldsymbol{x}_0^+$ and its covariance matrix, from which the initial latent coordinate $\boldsymbol{q}_0^+$ and covariance matrix $\mathbf{P}_0^+$ are then obtained through projection.

The left column of Figure 3 shows the ground truth state, the UKF posterior estimate of the full state and the corresponding estimation error, computed according to (46). As expected, the initial error is large because the GPR provides only a rough first guess. However, with successive corrections from the sensors, the error converges rapidly, within 5 t.u., to values below 0.1, demonstrating accurate state estimation of the 1D KS system.

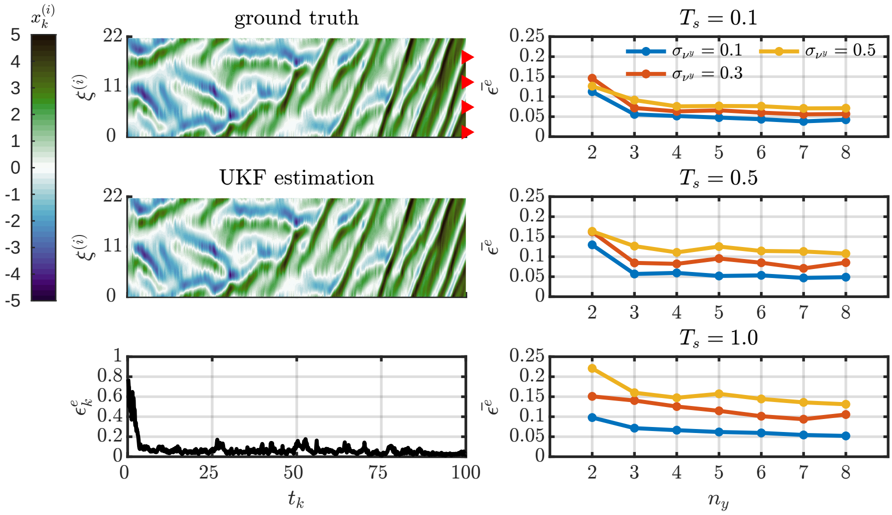

Figure 3: A posteriori Unscented Kalman Filter prediction of the state vector for controlled trajectories. Panels on the left show the ground truth of the state map and the a posteriori prediction for a case with 4 sensors (indicated by red triangles in the top figure), measurement noise intensity $\sigma_{\nu^y} = 0.1$, and measurement sampling period $T_s = 0.1$. Panels on the right show the average prediction error $\bar{\epsilon}^e$ over the time interval [50, 100], for different configurations of noise levels, number of sensors, and update frequencies. For each case, the initial condition of the prediction is estimated using a Gaussian Process Regression based on the sensor measurements available at the initial time step.

The estimation process for the 1D KS state is then repeated for different numbers of sensors ($n_y = 2, \dots, 8$), noise levels ($\sigma_{\nu^y} = 0.1,\ 0.3,\ 0.5$), and UKF correction intervals ($T_s = 0.1,\ 0.5,\ 1.0$ t.u.). As an indicator of estimation quality, the average value of the estimation error, denoted as $\bar{\epsilon}^e$, and calculated over the time window $t_k \in [50, 100]$, is computed for all cases using the same validation dataset. The results show a clear trend of decreasing estimation accuracy with fewer sensors, higher noise levels, and longer correction intervals.

3.3. *Set-point control*

Once the methodology for estimating the latent coordinate from sensor measurements at each time step has been established, output-feedback MPC can be applied. The initial estimate of the latent coordinate is used to evaluate the MPC cost function via model rollouts, as specified in (17).

The weight matrices of the MPC cost in (17) are selected as: $\mathbf{R}_u = 0.01\mathbf{I}_{n_u}$, and $\mathbf{R}_{\Delta u}$, $\mathbf{R}_{\Delta u} = 0.5\mathbf{I}_{n_u}$, to promote smooth control actions over time. Appropriate tuning of these matrices is crucial in problems involving multiple control targets; for instance, an automatic selection based on Bayesian optimisation is proposed in Marra *et al.* (2024) in the framework of aerodynamic forces minimisation for a wake flow. Here, however, the control objective is simpler, *i.e.* the stabilisation of the system around a target state, thus manual selection was chosen instead. The control inputs in the MPC are constrained to lie within the bounds $u_k^{(i)} \in [-10, 10]$. For the cases considered here, the prediction and control horizons are both set to $w_p = w_c = 2$ t.u.. Additionally, the terminal cost is estimated using the tail cost of a Linear Quadratic Regulator (LQR). Specifically, the OpInf model in (16) is linearised around the target state $\boldsymbol{q}^*$ with zero actuation, $\boldsymbol{u} = \mathbf{0}_{n_u}$, and the discrete-time Riccati equation is solved to obtain the terminal cost matrix $\mathbf{Q}_f$, so that the terminal cost in the MPC formulation is $V_f(\hat{\boldsymbol{q}}) = \hat{\boldsymbol{q}}^\top \mathbf{Q}_f \hat{\boldsymbol{q}}$. This ensures that the finite-horizon MPC captures the long-term effect of the control law beyond the prediction window.

To evaluate the control performance, the normalised RMSE is considered

$$\epsilon_k^c = \frac{\sqrt{\frac{1}{n_x} \sum_{i=1}^{n_x} \left(x_k^{(i)} - x_k^{*(i)} \right)^2}}{\sqrt{\frac{1}{n_x} \sum_{i=1}^{n_x} \left(x_k^{*(i)} \right)^2}}, \tag{47}$$

where $x_k^{(i)}$ and $x_k^{*(i)}$ denote the ith component of the true and target state vectors at time step t_k, respectively, and n_x is the dimension of the full-order state vector. This metric provides a time-step-wise measure of deviation from the desired target state, normalised by the root-mean-square of the target.

Figure 4 presents the results of the control around the three unstable invariant solutions, E_1, E_2, and E_3. For each case, an ensemble of 100 control trajectories is performed from random initial conditions. It is important to note that only partial measurements of the state are available, and the initial condition is estimated from the initial sensor measurements as in Figure 3. The UKF is run in a warm-up phase lasting 100 t.u., after which control is applied for 25 t.u..

For comparison, the figure also shows the ideal case of full observability, where the feedback consists of the complete, noise-free state of the 1D KS. A specific case with $n_y = 4$ sensors, measurement noise $\sigma_{\nu^y} = 0.1$, and sampling interval $T_s = 0.1$ is highlighted. The ensemble of states is plotted at two representative time instants, t_A and t_B, illustrating the control transients across all trajectories as they approach the target. In this scenario, the MPC successfully drives the true state to the target in all cases, with only minor discrepancies near E_2 attributed to local inaccuracies of the surrogate plant model rather than state estimation errors.

Beyond this case, control is tested for various configurations of sensor number ($n_y = 3, 4, 5, 6$), noise intensity ($\sigma_{\nu^y} = 0.1, 0.3, 0.5$), and sampling interval ($T_s = 0.1, 0.5$ t.u.). To compare the different cases, the ensemble average of the mean control error ($\bar{\epsilon}^c$) over the last 5 t.u. of the control horizon ($t_k \in [25, 30]$) is reported. The trend indicates that control performance decreases as the number of sensors decreases, or as measurement noise and sampling interval increase.

4. Scalability of the MPC framework to high-dimensional systems

This section demonstrates the extension of the proposed control framework to high-dimensional systems through its application to the two-dimensional Kuramoto-Sivashinsky (2D KS) equation. Let ξ and η denote the coordinates in the two spatial dimensions. The equation is defined on the rectangular domain $[0, 2L_\xi] \times [0, 2L_\eta]$, where L_ξ and L_η represent the half-lengths of the domain along each coordinate direction. The scalar field

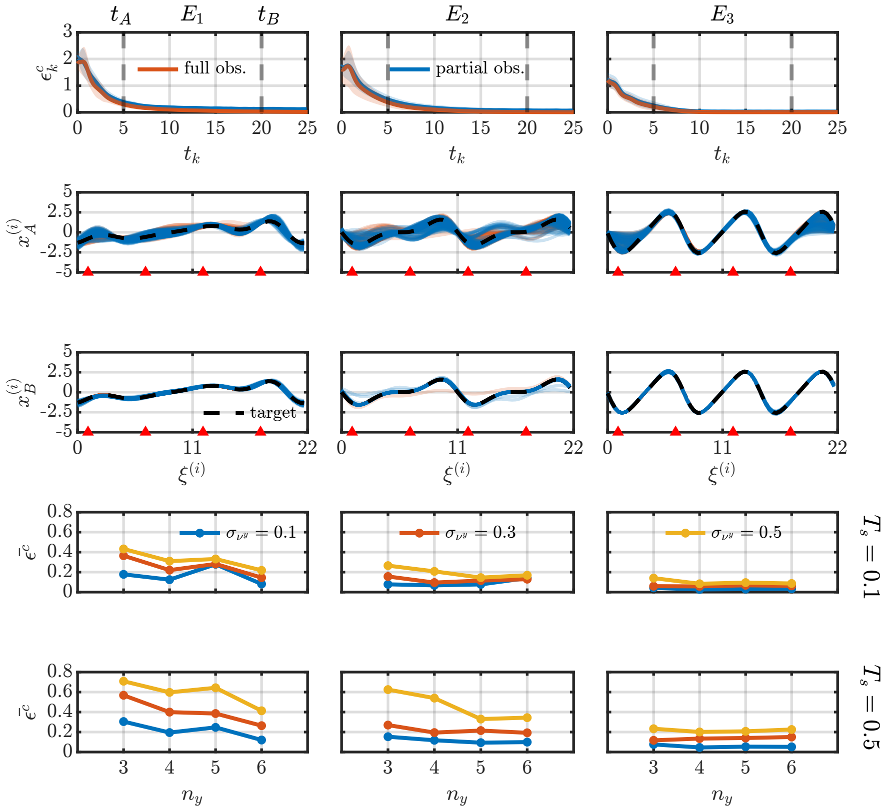

Figure 4: Results of the application of Model Predictive Control with an Unscented Kalman Filter state estimator. The control is performed around the three unstable equilibrium points of the 1D Kuramoto-Sivashinsky system, denoted as E_1 (left column), E_2 (middle column), and E_3 (right column). The first three rows correspond to a configuration with 4 sensors, measurement noise $\sigma_{\nu^y} = 0.1$, and sampling interval $T_s = 0.1$. The first row reports the control error of the true state with respect to the target (red solid line), while the blue line refers to the full-state feedback case for comparison. The second and third rows display the state evolution at two time instants, t_A and t_B, under partial observability (red) and full observability (blue). The last two rows present the ensemble-averaged error ($\bar{\epsilon^c}$) computed over the final 5 $t.u.$ of control.

$x(\xi, \eta, t)$ evolves over time according to

$$\frac{\partial x}{\partial t} + \frac{1}{2}\left|\nabla x\right|^2 + \Delta x + \Delta^2 x = 0, \tag{48}$$

where $\nabla = \left(\frac{\partial}{\partial \xi}, \frac{\partial}{\partial \eta}\right)^\top$ is the gradient operator, $\Delta = \frac{\partial^2}{\partial \xi^2} + \frac{\partial^2}{\partial \eta^2}$ denotes the Laplacian and $\Delta^2 = \Delta(\Delta)$ is the bi-Laplacian operator. The system is subject to periodic boundary conditions

$$x(\xi, \eta, t) = x(\xi + 2L_\xi, \eta, t) = x(\xi, \eta + 2L_\eta, t), \tag{49}$$

and the initial condition

$$x(\xi, \eta, 0) = x_0(\xi, \eta). \tag{50}$$

The solution of the KS equation $x(\xi, \eta, t)$ exhibits spatio-temporal dynamics arising from nonlinear advection (gradient term), linear diffusion (Laplacian term) and hyper-diffusion (bi-Laplacian).

For practical reasons, a rescaled version of the 2D KS equation, as presented in Kalogirou *et al.* (2015), is adopted through the transformation:

$$\xi \to \frac{L_\xi}{\pi}\xi, \quad \eta \to \frac{L_\eta}{\pi}\eta, \quad t \to \frac{L_\xi\, L_\eta}{\pi^2}t \tag{51}$$

which normalises the domain to $[0, 2\pi] \times [0, 2\pi]$. Under this rescaling, the equation becomes:

$$\frac{\partial x}{\partial t} + \frac{1}{2}\left|\nabla_\mu x\right|^2 + \Delta_\mu x + \mu_\xi \Delta_\mu^2 x = 0 \tag{52}$$

with

$$\nabla_\mu = \left(\frac{\partial}{\partial \xi}, \frac{\mu_\eta}{\mu_\xi}\frac{\partial}{\partial \eta}\right), \quad \Delta_\mu = \frac{\partial^2}{\partial \xi^2} + \frac{\mu_\eta}{\mu_\xi}\frac{\partial^2}{\partial \eta^2}, \tag{53}$$

and

$$\mu_\xi = \left(\frac{\pi}{L_\xi}\right)^2, \quad \mu_\eta = \left(\frac{\pi}{L_\eta}\right)^2. \tag{54}$$

The spatial mean, whose growth does not affect the higher modes, is defined as

$$\bar{x}(t) = \frac{1}{(2\pi)^2}\int_0^{2\pi}\int_0^{2\pi} x(\xi, \eta, t)\, d\xi\, d\eta \tag{55}$$

and is subtracted during the integration to enforce a mean-zero formulation.

In addition, to enable control, similarly to the one-dimensional case, the forcing term $\Gamma(\xi, \eta, t)$ is added in the right-hand side of (44). As done similarly in Jiang *et al.* (2025):

$$\Gamma(\xi, \eta, t) = \frac{1}{2\pi\sigma_a^2}\sum_{j=1}^{n_u} u^{(j)}(t)\, \exp\left(-\frac{(\xi - \xi_a^{(j)})^2 + (\eta - \eta_a^{(j)})^2}{2\sigma_a^2}\right), \tag{56}$$

where $u^{(j)}(t)$ is the time-dependent amplitude of the jth Gaussian actuator located at $(\xi_a^{(j)}, \eta_a^{(j)})$, the width is set to $\sigma_a = 0.8$ and the actuator centres are arranged on a 4×4 equispaced grid, yielding total of $n_u = 16$ actuators. The coordinates of the centres are $\{0, 1, 2, 3\}\frac{L_\xi}{4}$ and $\{0, 1, 2, 3\}\frac{L_\eta}{4}$ along the ξ and η coordinate respectively. Similarly, the sensors are assumed to be arranged in a regular grid with coordinates $\{1, \dots, n_y\}\frac{L_\xi}{n_y+1}$ and $\{1, \dots, n_y\}\frac{L_\eta}{n_y+1}$ along the two spatial dimensions. In addition, to monitor the behaviour of the solution, the energy of the state is defined as

$$E(t) = \int_0^{2\pi}\int_0^{2\pi} x^2(\xi, \eta, t)\ d\xi\ d\eta. \tag{57}$$

The equations are integrated using the spectral Galerkin solver *Shenfun*, a partial differential equations solver provided by Mortensen (2018). The equation is integrated on a computational grid of 32×32 points with an integration time step of $\Delta t = 0.0025$. This spatial and temporal resolution was found to be sufficient after comparison with two reference simulations: one using a 64×64 grid with the same integration time step, which showed solution differences of order $O(10^{-5})$, and another using a 32×32 grid with a smaller time step of $\Delta t = 0.001$, which exhibited errors of order $O(10^{-3})$.

The dynamics of the equation are determined by the selection of the scaling factor of the domain μ_ξ and μ_η, ranging from steady solutions to chaos. A study of the behaviour of the 2D KS equation for different realisations of the scaling parameter is proposed in Kalogirou *et al.* (2015). The parameters selected in this case are equal to $\mu_\xi = 0.3$ and $\mu_\eta = 0.16$. Under this selection, the system exhibits a modulated periodic regime, alternating between stable travelling waves and oscillatory phases with repeated, identical amplitude patterns.

A dataset is first generated, after which the OpInf procedure is applied. The time step for model identification and subsequent control of the system is set to $\Delta t_c = 0.02$. The total duration of the training dataset is chosen as $T_{\mathrm{tr}} = 3000$ t.u., yielding $M = 150{,}000$ snapshots. The dataset includes the first 500 t.u. of the unforced solution, followed by a controlled solution with Fourier-filtered Gaussian actuations with a frequency cut-off $f_{co} = 2$. After the frequency filter, the control actions are centred to have zero mean and standard deviation $\sigma_u = 3$.

A representation of the training dataset is included in Figure 5, corresponding to the time interval $t_k \in [450, 550]$. The figure highlights both the free-evolution (green shaded region) and the forced segment. Snapshots of the system states at selected times during the unforced phase (t_A, t_B, t_C, and t_D) and during the forced phase (t_E and t_F) are displayed alongside the corresponding control fields. In the uncontrolled case, intervals with nearly constant solution energy correspond to travelling waves (as seen in the snapshots at t_C and t_D), where the solution translates across the domain at a speed determined by the wave. The figure also illustrates the oscillatory pattern that arises when the travelling wave becomes unstable. This pattern repeats periodically in time and smoothly evolves from the preceding travelling wave solution, as shown at t_A and t_B. The figure also shows the configuration of the actuations. At time instant t_E, the positions of the actuator centres are indicated by yellow cross points, and the contour corresponding to one standard deviation of the Gaussian actuators is also shown.

Using the training dataset described above, the OpInf method is applied to learn a surrogate model for the 2D KS system. The number of retained modes for the low-dimensional reconstruction is determined based on the variance captured, with $\gamma_r = 0.9999$, resulting in a reduced space of dimension $r = 60$. The model as described by (9) includes the constant term $\mathbf{c}$, the linear state operator $\mathbf{A}$, the quadratic state operator $\mathbf{H}$, and the linear input operator $\mathbf{B}$. Regularisation for the optimisation of the operator coefficients is set to $\lambda_o = 0.1$, following the same approach used for the 1D KS case. The resulting surrogate model, including quadratic polynomial terms, is then used in a data assimilation process where sparse, noisy measurements of the state provide feedback in the UKF framework. The identified model exhibits a relative one-step prediction bias of 0.007.

Figure 6 shows the results of the UKF-based state estimation for both uncontrolled and controlled scenarios. The covariance matrices of the process and measurement noise are fixed to $\mathbf{Q} = 0.01\mathbf{I}_{n_q}$ and $\mathbf{R} = \sigma_{\nu^y}^2 \mathbf{I}_{n_q}$. The unscented transformation in UKF is again configured with the parameters $\alpha = 0.1$, $\beta = 2.0$, and $\kappa = 0$. The initial condition is assumed unknown and estimated via GPR using the available sensor measurements at the initial time. In the left and centre columns of the figure, the UKF is tested using a regular 4×4 sensor grid, measurement noise level $\sigma_{\nu^y} = 0.1$, and a sampling interval of $T_s = 0.02$.

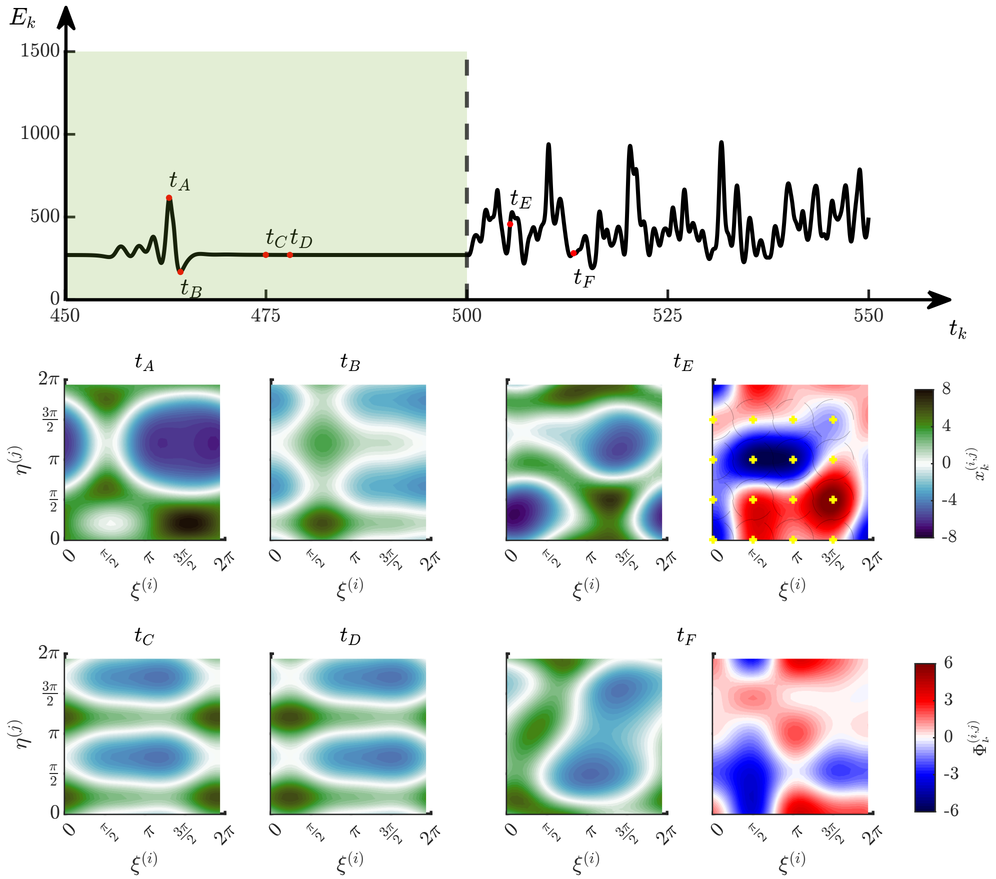

Figure 5: Representation of a subset of the training dataset used for system identification in the 2D Kuramoto–Sivashinsky case. The central panel depicts the time evolution of the state energy over the interval $t_k \in [450, 550]$. The green-shaded region corresponds to uncontrolled dynamics, while the unshaded region includes random, smooth actuation. The top and bottom panels display the state field (blue–green colourmap) at selected time instants within the uncontrolled (t_A, t_B, t_C, t_D) and controlled (t_E, t_F) regions. For times t_D and t_F, the corresponding actuation maps (blue–red colourmap) are also shown. Yellow crosses indicate the actuator centres, and the contour of each actuator region corresponding to one standard deviation of the Gaussian is shown.

The left panels correspond to the uncontrolled case, while the central panels show the controlled scenario, using actuation patterns similar to those employed in generating the training dataset. For both cases, the figure reports the temporal evolution of the state energy E_k in a time range $t_k \in [0, 20]$, the estimation error maps at three representative time instants (t_A, t_B, t_C), and the locations of the sensors (red dots). The estimation error, defined in (46) and averaged over $t_k \in [10, 20]$, is evaluated for different sensor configurations (from 3×3 to 5×5 grids), noise levels ($\sigma_{\nu^y} = 0.1, 0.3, 0.5$), and sampling intervals ($T_s = 0.02, 0.1$). The curves for $\sigma_{\nu^y} = 0.1$, 0.3, and 0.5 are shown in blue, red,

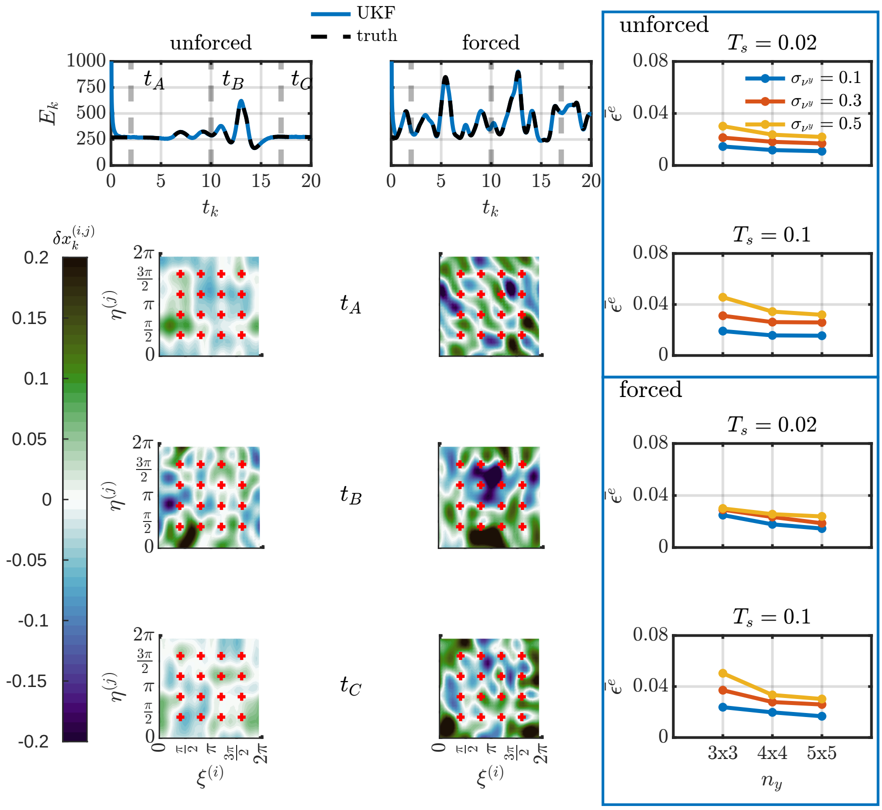

Figure 6: A posteriori UKF estimation of the state field for the 2D Kuramoto–Sivashinsky system in controlled and uncontrolled conditions. The left panels correspond to an uncontrolled case with a 4×4 sensor grid, measurement noise $\sigma_{\nu^y} = 0.1$, and sampling interval $T_s = 0.1$, while the central panels show the corresponding controlled case with random smooth actuation. Red dots indicate the sensor locations for the depicted 4×4 configuration. For both cases, the first row presents the temporal evolution of the state energy E_k in the time interval $t_k \in [0, 20]$ for the ground truth and the Unscented Kalman Filter estimate, followed by the estimation error maps at three time instants, t_A, t_B, and t_C. The right panels report the average estimation error ($\bar{\epsilon^e}$) over the final 10 $t.u.$ of the trajectories, for different configurations of sensor grids, measurement noise levels, and sampling intervals.

and yellow, respectively. In all tested cases, the estimation error remains below $\approx 5\%$. Notably, the 4×4 sensor configuration achieves estimation accuracy comparable to the 5×5 grid, indicating that further increasing the number of sensors provides only marginal improvements.

The learned model is subsequently employed within the MPC framework. The control objective is formulated as a set-point regulation problem, where the target state corresponds to the steady-state solution observed for $\mu_\xi = \mu_\eta = 0.9$. The weighting matrices in the cost function are chosen as $\mathbf{R}_q = \mathbf{I}_{n_q}$, $\mathbf{R}_u = 0.01\mathbf{I}_{n_u}$, and $\mathbf{R}_{\Delta u} = 0.5\mathbf{I}_{n_u}$. The control inputs are constrained within $u_k^{(i)} \in [-15, 15]$, and both the prediction and control horizons are set to $w_p = w_c = 1.5$ t.u..

Before addressing the partially-observable control case, the performance of the learned surrogate model is first assessed under full-state feedback conditions. In this configuration, the latent state is directly available to the controller at each step. Figure 7 summarises the results for an ensemble of 100 control trajectories, each 10 t.u. long and initialised from distinct conditions to assess generalisability. The top panel shows the control sequence for a representative trajectory corresponding to the median control error within the ensemble. The evolution of the 16 actuator amplitudes (weights of the Gaussian actuators) reveals an initial coordination phase that drives the solution towards the steady-state target. The subsequent panels display the controlled state fields and actuation maps at times t_A, t_B, and t_C. The bottom panels of the figure report the target field (left) and the ensemble-averaged control error (right), as defined in Eq. (47), together with its $\pm\sigma$ confidence interval. Across the ensemble, the mean control error stabilises around 19%, confirming that the OpInf-based model provides reliable performance for state regulation under full observability.

Finally, the partially observed control scenario is presented in Figure 8. Here, the feedback to the MPC is provided by the state estimate obtained from the UKF using sparse and noisy measurements. The study explores the effect of varying the number of sensors (from 3×3 to 5×5 grids), the measurement noise level ($\sigma_{\nu^y} = 0.1, 0.3, 0.5$), and the sampling interval ($T_s = 0.02, 0.1$). For each configuration, an ensemble of 20 control trajectories with distinct initial conditions is simulated. The control performance is quantified using the time-averaged error over the final 2 t.u. of each trajectory, and averaged across the ensemble. The results show that increasing the number of sensors consistently improves both the estimation and control accuracy, with the 5×5 configuration outperforming the sparser grids in all tested scenarios. A representative trajectory for the case with a 4×4 sensor grid, $\sigma_{\nu^y} = 0.1$, and $T_s = 0.02$ is also displayed in the figure. The top row presents the temporal evolution of the actuation history, followed by the true and UKF-estimated state fields at three control instants (t_A, t_B, t_C), and the corresponding actuation maps. The rightmost panels summarise the target state, the ensemble-averaged control error, and the averaged estimation error for all sensor and noise configurations. Overall, these results demonstrate that the combination of OpInf-based model reduction, UKF estimation, and MPC control achieves good performance even under sparse and noisy partial observations.

5. Conclusion

This work proposed a model predictive control framework tailored for high-dimensional nonlinear systems with complex spatio-temporal dynamics. The framework directly tackles the challenge of the scalability of model predictive control in high-dimensional systems with limited observability. The proposed approach integrates reduced-order modelling, state estimation, and control within a consistent latent-space formulation. The predictive model is obtained via Operator Inference on a Proper Orthogonal Decomposition basis, yielding a compact, data-driven surrogate that captures the essential nonlinear behaviour of the original system. State feedback is achieved through an Unscented Kalman Filter that estimates the latent state from sparse and noisy measurements, ensuring accurate reconstruction despite limited sensing.

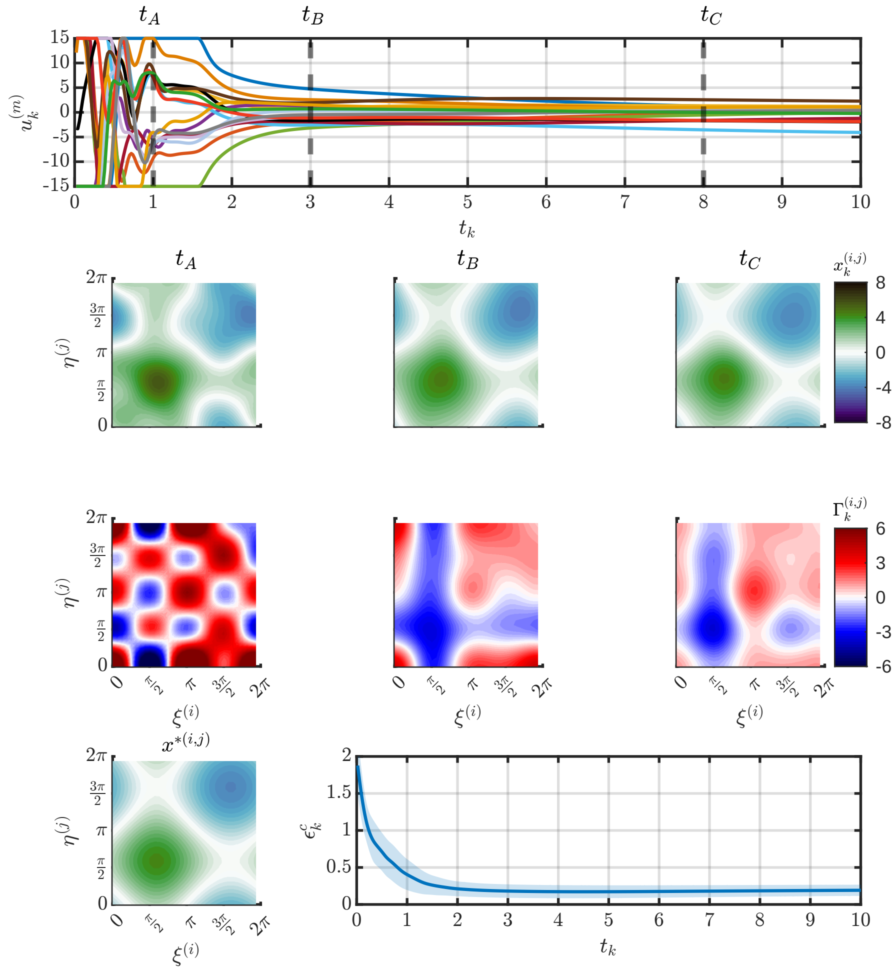

Figure 7: Control of the 2D-KS system under full observability, where the full state is provided as control feedback. The top section illustrates the actuation history for a median-error control trajectory (top row), the corresponding state fields (second row), and the actuation maps (third row) at three different time instants, t_A, t_B, and t_C. The bottom panels display the target state (left) and the average control error (blue solid line) with its $\pm\sigma$ confidence region (shaded blue area) computed over an ensemble of 100 control trajectories.

The method has been validated on one- and two-dimensional Kuramoto–Sivashinsky systems, demonstrating effective closed-loop stabilisation and reliable estimation performance. These results highlight the potential of the proposed approach as a tractable and general strategy for feedback control of complex, high-dimensional systems like the ones arising in fluid dynamics.

It is thus demonstrated that integration of an unscented Kalman Filter estimator within the framework of model predictive control based on latent-space formulations

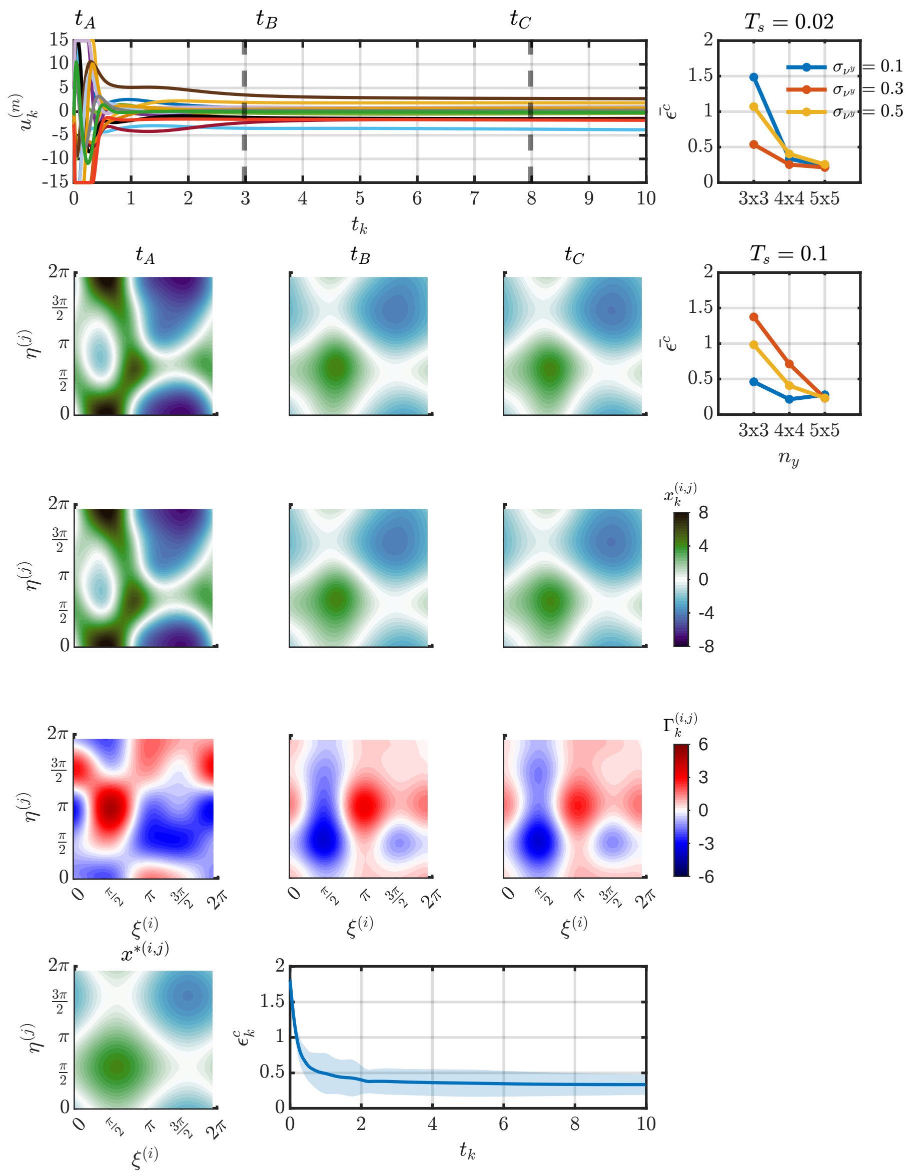

Figure 8: Control of the 2D Kuramoto–Sivashinsky system under partial observability. The top row shows the actuation history over time for a representative median error control trajectory. The second row displays the true state at three selected control instants, t_A, t_B, and t_C, while the third row shows the corresponding estimated states using the Unscented Kalman Filter. The fourth row presents the MPC-computed actuation maps at the same instants. The bottom panels summarise the target state and the control error over time for an ensemble of 20 trajectories. The ensemble-averaged estimation error ($\bar{\epsilon^c}$) over the final 2 *t.u.* of control, for different sensor numbers, noise levels, and sampling intervals, is shown in the rightmost column.

enables robust kick-off of the control by progressive regularisation of the full-state estimate, and proper steering of the predictive model to avoid error divergence due to truncation of the dynamics of modelling inaccuracies in the latent space. This approach is envisioned as a solid framework applicable to fields that are severely affected by both limited observability and high dimensionality, which are ubiquitous in fluid dynamics, chemistry, biology and climate science. From this perspective, the proposed methodology has to be viewed as an alternative to standard control strategies, or to reinforcement learning, where partial observability is typically addressed through memory-based mechanisms such as recurrent neural networks or attention-based architectures (Weissenbacher *et al.*, 2025). Unlike memory-dependent Reinforcement Learning approaches, this framework uses an explicit state-estimation procedure, thereby avoiding issues of long-term error accumulation. This highlights the potential of latent-space, data-driven control methods as a complementary approach for controlling complex chaotic systems under limited sensing, balancing between interpretability and computational efficiency.

Future work will focus on the development of robust model predictive control extensions, quantifying modelling and estimation errors. Further investigations will target applications to more challenging fluid-dynamics configurations. Additionally, alternative latent-space representations, such as neural-network-based autoencoders, will be explored to improve model expressiveness and scalability.

References

Agostini, Lionel 2020 Exploration and prediction of fluid dynamical systems using auto-encoder technology. *Phys. Fluids.* **32** (6), 067103.

Allgower, F., Findeisen, R., Nagy, Z. K. & others 2004 Nonlinear model predictive control: From theory to application. *J. Chin. Inst. Chem. Eng.* **35** (3), 299–316.

Alora, J. I., Pabon, L. A., Köhler, J., Cenedese, M., Schmerling, E., Zeilinger, M. N., Haller, G. & Pavone, M. 2023 Robust nonlinear reduced-order model predictive control. In *2023 62nd IEEE Conference on Decision and Control (CDC)*, pp. 4798–4805.

Amendola, M., Arcucci, R., Mottet, L., Casas, C. Q., Fan, S., Pain, C., Linden, P. & Guo, Y.-K. 2021 Data assimilation in the latent space of a convolutional autoencoder. In *ICCS*, pp. 373–386. Springer.

Andersson, J. A. E., Gillis, J., Horn, G., Rawlings, J. B. & Diehl, M. 2019 CasADi – A software framework for nonlinear optimization and optimal control. *Math. Program. Comput.* **11** (1), 1–36.

Arbabi, H., Korda, M. & Mezić, I. 2018 A data-driven koopman model predictive control framework for nonlinear partial differential equations. In *2018 IEEE Conference on Decision and Control (CDC)*, pp. 6409–6414. IEEE.

Bertsekas, D. P. 2024 Model predictive control and reinforcement learning: A unified framework based on dynamic programming. *IFAC-PapersOnLine* **58** (18), 363–383, 8th IFAC Conference on Nonlinear Model Predictive Control NMPC 2024.

Bewley, T. R., Moin, P. & Temam, R. 2001 Dns-based predictive control of turbulence: an optimal benchmark for feedback algorithms. *J. Fluid Mech.* **447**, 179–225.

Bieker, K., Peitz, S., Brunton, S. L., Kutz, J. N. & Dellnitz, M. 2020 Deep model predictive flow control with limited sensor data and online learning. *Theor. Comput. Fluid Dyn.* **34** (4), 577–591.

Billings, S. A. 2013 *Nonlinear system identification: NARMAX methods in the time, frequency, and spatio-temporal domains*. John Wiley & Sons.

Box, G.E.P., Jenkins, G. M., Reinsel, G. C. & Ljung, G. M. 2015 *Time series analysis: forecasting and control*. John Wiley & Sons.

Brindise, M. C. & Vlachos, P. P. 2017 Proper orthogonal decomposition truncation method for data denoising and order reduction. *Exp Fluids.* **58** (4), 28.

Brunton, S. L. & Kutz, J. N. 2022 *Data-driven science and engineering: Machine learning, dynamical systems, and control*. Cambridge University Press.

Brunton, S. L. & Noack, B. R. 2015 Closed-loop turbulence control: Progress and challenges. *Appl. Mech. Rev.* **67** (5), 050801.

Brunton, S. L., Noack, B. R. & Koumoutsakos, P. 2020 Machine learning for fluid mechanics. *Annu. Rev. Fluid Mech.* **52** (Volume 52, 2020), 477–508.

Brunton, S. L., Proctor, J. L. & Kutz, J. N. 2016*a* Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *PNAS* **113** (15), 3932–3937.

Brunton, S. L., Proctor, J. L. & Kutz, J. N. 2016*b* Sparse identification of nonlinear dynamics with control (sindyc). *IFAC-PapersOnLine* **49** (18), 710–715.

Bucci, M. A., Semeraro, O., Allauzen, A., Wisniewski, G., Cordier, L. & Mathelin, L. 2019 Control of chaotic systems by deep reinforcement learning. *Proc. R. Soc. A* **475** (2231), 20190351.

Camacho, E. F. & Bordons, C. 2007 Constrained model predictive control. In *Model predictive control*, pp. 177–216. Springer.

Chiuso, A. & Pillonetto, G. 2019 System identification: A machine learning perspective. *Annu. Rev. Control Robot.* **2** (Volume 2, 2019), 281–304.

Dankowicz, H., Holmes, P., Berkooz, G. & Elezgaray, J. 1996 Local models of spatio-temporally complex fields. *Physica D: Nonlinear Phenomena* **90** (4), 387–407.

Dee, D. P. 2005 Bias and data assimilation. *Q. J. Roy. Meteor. Soc.* **131** (613), 3323–3343.

Déda, T., Wolf, W. R. & Dawson, S. T. M. 2023 Backpropagation of neural network dynamical models applied to flow control. *Theor. Comput. Fluid Dyn.* **37** (1), 35–59.

Evensen, G. 2003 The ensemble kalman filter: Theoretical formulation and practical implementation. *Ocean Dyn.* **53** (4), 343–367.

Fabbiane, N., Semeraro, O., Bagheri, S. & Henningson, D. S. 2014 Adaptive and model-based control theory applied to convectively unstable flows. *Appl. Mech. Rev.* **66** (6), 060801–1–060801–20.

Findeisen, R., Imsland, L., Allgower, F. & Foss, B. A. 2003 State and output feedback nonlinear model predictive control: An overview. *Eur. J. Control.* **9** (2-3), 190–206.

Hovland, S., Willcox, K. & Gravdahl, J. T. 2006 Mpc for large-scale systems via model reduction and multiparametric quadratic programming. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pp. 3418–3423.

Janiszewski, D. 2024 Sensorless model predictive control of permanent magnet synchronous motors using an unscented kalman filter. *Energies* **17** (10).

Jiang, J., Wan, D. & Zhang, M. 2025 Numerical solutions of fixed points in two-dimensional kuramoto–sivashinsky equation expedited by reinforcement learning. *Proc. R. Soc. A* **481** (2306), 20240222.

Jiang, Z., Song, Q., He, Y. & Han, J. 2007 A novel adaptive unscented kalman filter for nonlinear estimation. In *2007 46th IEEE Conference on Decision and Control*, pp. 4293–4298. IEEE.

Julier, S. J. & Uhlmann, J. K. 1997 New extension of the kalman filter to nonlinear systems. In *Signal processing, sensor fusion, and target recognition VI*, , vol. 3068, pp. 182–193. Spie.

Kalogirou, A., Keaveny, E. E. & Papageorgiou, D. T. 2015 An in-depth numerical study of the two-dimensional kuramoto–sivashinsky equation. *Proc. R. Soc. A* **471** (2179), 20140932.

Karg, B. & Lucia, S. 2021 Approximate moving horizon estimation and robust nonlinear model predictive control via deep learning. *Comput. Chem. Eng.* **148**, 107266.

Kramer, B., Peherstorfer, B. & Willcox, K. E. 2024 Learning nonlinear reduced models from data with operator inference. *Annu. Rev. Fluid Mech.* **56**, 521–548.

Kuramoto, Y. 1978 Diffusion-induced chaos in reaction systems. *Prog. Theor. Phys. Suppl.* **64**, 346–367.

Lee, J. H. & Ricker, N. L. 1994 Extended kalman filter based nonlinear model predictive control. *Ind. Eng. Chem. Res.* **33** (6), 1530–1541.

Ljung, L. 2003 Convergence analysis of parametric identification methods. *IEEE Trans. Autom. Control.* **23** (5), 770–783.

Lu, Q. & Zavala, V. M. 2021 Image-based model predictive control via dynamic mode decomposition. *J. Process Control* **104**, 146–157.

Marafioti, G., Olaru, S. & Hovd, M. 2009 *State Estimation in Nonlinear Model Predictive Control, Unscented Kalman Filter Advantages*, pp. 305–313. Berlin, Heidelberg: Springer Berlin Heidelberg.

Marra, L., Meilán-Vila, A. & Discetti, S. 2024 Self-tuning model predictive control for wake flows. *J. Fluid Mech.* **983**, A26.

Mayne, D. Q., Raković, S.V., Findeisen, R. & Allgöwer, F. 2006 Robust output feedback model predictive control of constrained linear systems. *Automatica* **42** (7), 1217–1222.

Mortensen, M. 2018 Shenfun: High performance spectral galerkin computing platform **3** (31), 1071.

Morton, J., Jameson, A., Kochenderfer, M. J. & Witherden, F. 2018 Deep dynamical modeling and control of unsteady fluid flows. *NeurIPS* **31**.

Muske, K. R. & Rawlings, J. B. 1993 Model predictive control with linear models. *AIChE journal* **39** (2), 262–287.

Nóvoa, A. & Magri, L. 2025 Online model learning with data-assimilated reservoir computers. In *Comput. Sci – ICCS 2025*, pp. 57–64. Cham: Springer Nature Switzerland.

Nóvoa, A., Racca, A. & Magri, L. 2024 Inferring unknown unknowns: Regularized bias-aware ensemble kalman filter. *Comput. Methods Appl. Mech. Eng.* **418**, 116502.

Özalp, E., Margazoglou, G. & Magri, L. 2023 Reconstruction, forecasting, and stability of chaotic dynamics from partial data. *Chaos* **33** (9), 093107.

Özalp, E., Nóvoa, A. & Magri, L. 2026 Real-time forecasting of chaotic dynamics from sparse data and autoencoders. *Comput. Methods Appl. Mech. Eng.* **450**, 118600.

Ozan, D. E., Nóvoa, A., Rigas, G. & Magri, L. 2025 Data-assimilated model-informed reinforcement learning. *arXiv preprint* ArXiv:2506.01755.

Papageorgiou, D. T. & Smyrlis, Y. S. 1991 The route to chaos for the kuramoto-sivashinsky equation. *Theor. Comput. Fluid Dyn* **3** (1), 15–42.

Peherstorfer, B. & Willcox, K. 2016 Data-driven operator inference for nonintrusive projection-based model reduction. *Comput. Methods Appl. Mech. Eng.* **306**, 196–215.

Peyron, M., Fillion, A., Gürol, S., Marchais, V., Gratton, S., Boudier, P. & Goret, G. 2021 Latent space data assimilation by using deep learning. *Q. J. R. Meteorol. Soc.* **147** (740), 3759–3777.

Qian, E., Farcas, I.-G. & Willcox, K. 2022 Reduced operator inference for nonlinear partial differential equations. *SIAM J. Sci. Comput.* **44** (4), A1934–A1959.

Racca, A., Doan, N. A. K. & Magri, L. 2023 Predicting turbulent dynamics with the convolutional autoencoder echo state network. *J. Fluid Mech.* **975**, A2.

Raissi, M. & Karniadakis, G. E. 2018 Hidden physics models: Machine learning of nonlinear partial differential equations. *J. Comput. Phys.* **357**, 125–141.

Rempfer, D 2000 On low-dimensional galerkin models for fluid flow. *Theor. Comput. Fluid Dyn.* **14** (2), 75–88.

Ribeiro, M. I. 2004 Kalman and extended kalman filters: Concept, derivation and properties. *ISR* **43** (46), 3736–3741.

Sangiorgio, M., Dercole, F. & Guariso, G. 2021 Forecasting of noisy chaotic systems with deep neural networks. *Chaos, Solitons & Fract.* **153**, 111570.

Scardua, L. A. & da Cruz, J. J. 2017 Complete offline tuning of the unscented kalman filter. *Automatica* **80**, 54–61.

Schulz, E., Speekenbrink, M. & Krause, A. 2018 A tutorial on gaussian process regression: Modelling, exploring, and exploiting functions. *J. Math. Psychol.* **85**, 1–16.

Schüssler, M., Münker, T. & Nelles, O. 2019 Deep recurrent neural networks for nonlinear system identification. In *2019 IEEE symposium series on computational intelligence (SSCI)*, pp. 448–454. IEEE.

Seeger, M. 2004 Gaussian processes for machine learning. *Int. J. Neural Syst.* **14** (02), 69–106, pMID: 15112367.

Siv Ashinsky, G.I. 1988 Nonlinear analysis of hydrodynamic instability in laminar flames—i. derivation of basic equations. In *Dynamics of Curved Fronts* (ed. Pierre Pelcé), pp. 459–488. San Diego: Academic Press.

Sivashinsky, G. I. 1980 On flame propagation under conditions of stoichiometry. *SIAM J. Appl. Math.* **39** (1), 67–82.

Tenny, M. J. & Rawlings, J. B. 2002 Efficient moving horizon estimation and nonlinear model predictive control. In *Proceedings of the 2002 American Control Conference (IEEE Cat. No. CH37301)*, , vol. 6, pp. 4475–4480. IEEE.

Wächter, A. & Biegler, L. T. 2006 On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Math. Program.* **106** (1), 25–57.

Wan, E. A. & Van Der Merwe, R. 2000 The unscented kalman filter for nonlinear estimation. *Proceedings of the IEEE Adaptive Systems for Signal Processing, Communications, and Control Symposium* pp. 153–158.

Weissenbacher, M., Borovykh, A. & Rigas, G. 2025 Reinforcement learning of chaotic systems control in partially observable environments. *Flow Turbul. Combust.* pp. 1–22.

Welch, G. & Bishop, G. 1995 An introduction to the kalman filter. *UNC-Chapel Hill* .

Zhang, M., Zeng, F., Yousif, M. Z., Zhu, Z., Lee, J. S. & Lim, H.-C. 2025 Convolutional autoencoder-augmented sparse identification of noisy nonlinear fluid dynamics from unlabelled data. *Phys. Fluids* **37** (9), 095177.

Zhang, Y., Li, M., Zhang, Y., Hu, Z., Sun, Q. & Lu, B. 2022 An enhanced adaptive unscented kalman filter for vehicle state estimation. *IEEE Transactions on Instrumentation and Measurement* **71**, 1–12.

Paper 3

Actuation manifold from snapshot data

L. Marra[1], G. Y. Cornejo Maceda[2], A. Meilán-Vila[3], V. Guerrero[3], S. Rashwan[1], B. R. Noack[2,4], S. Discetti[1], & A. Ianiro[1]

[1] Department of Aerospace Engineering, Universidad Carlos III de Madrid, Av. de la Universidad 30, Leganés, 28911, Madrid, Spain
[2] Chair of Artificial Intelligence and Aerodynamics, School of Mechanical Engineering and Automation, Harbin Institute of Technology, 518055 Shenzhen, P. R. China [3] Department of Statistics, Universidad Carlos III de Madrid, Calle Madrid, 126, Getafe, 28903, Madrid, Spain [4] Guangdong Provincial Key Laboratory of Intelligent Morphing Mechanisms and Adaptive Robotics, Harbin Institute of Technology, 518055 Shenzhen, P. R. China

Journal of Fluid Mechanics (2024), vol. **996**, A26. DOI:
`https://doi.org/10.1017/jfm.2024.593`

We propose a data-driven methodology to learn a low-dimensional manifold of controlled flows. The starting point is resolving snapshot flow data for a representative ensemble of actuations. Key enablers for the actuation manifold are isometric mapping as encoder, and a combination of a neural network and a k-nearest-neighbour interpolation as decoder. This methodology is tested for the fluidic pinball, a cluster of three parallel cylinders perpendicular to the oncoming uniform flow. The centres of these cylinders are the vertices of an equilateral triangle pointing upstream. The flow is manipulated by constant rotation of the cylinders, i.e. described by three actuation parameters. The Reynolds number based on a cylinder diameter is chosen to be 30. The unforced flow yields statistically symmetric periodic shedding represented by a one-dimensional limit cycle. The proposed methodology yields a five-dimensional manifold describing a wide range of dynamics with small representation error. Interestingly, the manifold coordinates automatically unveil physically meaningful parameters. Two of them describe the downstream periodic vortex shedding. The other three describe the near-field actuation, i.e. the strength of boat-tailing, the Magnus effect and forward stagnation point. The manifold is shown to be a key enabler for control-oriented flow estimation.

Key words: Flow Control, Low-Dimensional Models, Machine Learning

---

1. Introduction

In this study, we propose a data-driven manifold learner of the flows for a large range of operating conditions demonstrated by a low-dimensional manifold for the actuated fluidic pinball. A cornerstone of theoretical fluid mechanics is the low-dimensional representation of coherent structures. This representation is a basis for understanding dynamic modelling, sensor-based flow estimation, model-based control, and optimisation.

Many avenues of reduced-order representations have been proposed. Leonardo da Vinci presented coherent structures and vortices artfully as sketches in a time when the Navier-Stokes equations were not known (Marusic & Broomhall, 2021). A quantitative pathway of low-dimensional modelling was started by Hermann von Helmholtz (1858) with his vortex laws. Highlights are the von Kármán (1912) model of the vortex street and derivation of feedback wake stabilization from the Föppl's vortex model by Protas (2004). Orr and Sommerfeld added a stability framework culminating in Stuart's mean-field theory (Stuart, 1958) to incorporate non-linear Reynolds stress effects. The proper orthogonal decomposition (POD) has quickly become the major data-driven approach to compress snapshot data into a low-dimensional Galerkin expansion (Berkooz *et al.*, 1993). Cluster-based modelling is an alternative data-driven flow compression, coarse-graining snapshot data into a small set of representative centroids (Kaiser *et al.*, 2014). All these low-dimensional flow representations are the kinematic prelude to dynamic models, e.g. point vortex models, modal stability approaches (Theofilis, 2011), POD Galerkin models (Holmes *et al.*, 2012) and cluster-based network models (Fernex *et al.*, 2021).

Data-driven manifold learners are a recent highly promising avenue of reduced-order representations. A two-dimensional manifold may, for instance, accurately resolve transient cylinder wakes: the manifold dimension is a tiny fraction of the number of vortices, POD modes and clusters required for a similar resolution. This manifold may be obtained by mean-field considerations (Noack *et al.*, 2003), by dynamic features (Loiseau *et al.*, 2018), by locally linear embedding (Noack *et al.*, 2023) and by isometric mapping (ISOMAP, Farzamnik *et al.*, 2023). In all these approaches, the manifolds have been determined for a single operating condition. Haller *et al.* (2023) emphasised a key challenge, namely, it remains *"unclear if these manifolds are robust under parameter changes or under the addition of external, time-dependent forcing".* In this study, we thus aim to identify a manifold representing fluid flows for a rich set of control actions by incorporating the actuation commands.

We choose the fluidic pinball as our benchmark problem for the proposed manifold learning. This configuration has been proposed by Ishar *et al.* (2019) as a geometrically simple two-dimensional configuration with a rich dynamic complexity under cylinder actuation and Reynolds number change. The transition scenario is described and modelled by Deng *et al.* (2020) comprising a sequence of bifurcations before passing to chaotic behaviour. The feedback stabilisation achieved through cylinder rotation has been accomplished with many different approaches (Raibaudo *et al.*, 2020; Cornejo Maceda *et al.*, 2021; Li *et al.*, 2022; Wang *et al.*, 2023). Farzamnik *et al.* (2023) have demonstrated that the unforced dynamics can be described on a low-dimensional manifold.

In this study, we illustrate the development of the control-oriented manifold — called an 'actuation manifold' in the following — for the fluidic pinball with three steady cylinder rotations as independent inputs. ISOMAP is chosen for its superior compression capability and manifold interpretability over other techniques such as POD (Farzamnik *et al.*, 2023).

The rest of the article is structured as follows. In §2 we introduce the dataset of the actuated fluidic pinball while in §3 we discuss the methodology employed to distill the actuation manifold and to use it for flow-estimation purposes. The results are presented and discussed in §4, before highlighting the main conclusions of the work in §5.

2. Flow control dataset

The dataset is based on two-dimensional incompressible uniform flow past three cylinders of equal diameter D. Their vertices form an equilateral triangle with one vertex pointing upstream and its median aligned with the streamwise direction. The origin of the

Cartesian reference system is located in the midpoint between the two rightmost cylinders. The streamwise and crosswise directions are indicated with x and y, respectively. The corresponding velocity components are indicated with u and v. The computational domain Ω is bounded in $[-6, 20] \times [-6, 6]$, and the unstructured grid used has 4225 triangles and 8633 nodes. The snapshots are linearly interpolated on a structured grid with a spacing of 0.05 in both x and y directions to maintain a format similar to the experimental data and for greater simplicity.

The Reynolds number (Re), which is based on D and the incoming velocity U_∞, is set to 30 for all cases. A reference time scale D/U_∞ is set as a convective unit (c.u.). The force coefficients C_L and C_D are obtained by normalising respectively lift and the drag forces with $\frac{1}{2}\rho U_\infty^2 D$, being ρ the density of the fluid.

The control is achieved with independent cylinder rotations included in the vector $\boldsymbol{b} = (b_1, b_2, b_3)^T$, where b_1, b_2 and b_3 refer to the tangential speeds of the front, top and bottom cylinder, respectively. Here, 'T' denotes the transpose operator. Positive actuation values correspond to counter-clockwise rotations. Combinations of rotational speed ranging from -3 to 3 by steps of 1 (i.e., $-3, -2, \dots, 3$) have been used, for a total of 343 configurations, including the unforced case. Each of these simulation is run for 800 c.u. to reach the steady state and snapshots with 1 c.u. separation in the last 20 c.u. for each steady state have been ¡sampled. This time is chosen to approximately cover two shedding cycles of the unforced case. This dataset therefore consists of 20 post-transient snapshots for each of the 343 combinations of actuations explored, for a total of $M = 6860$ snapshots.

The actuations are condensed into a three-parameter vector, denoted as $\boldsymbol{p} = (p_1, p_2, p_3)^T$, and referred to as Kiki parameters (Lin, 2021). These three parameters describe the three mechanisms of boat-tailing ($p_1 = (b_3 - b_2)/2$), Magnus effect ($p_2 = b_1 + b_2 + b_3$) and forward stagnation point control ($p_3 = b_1$). The Kiki parameters are introduced for an easier physical interpretation of the actuation commands.

3. Methodology

A n-dimensional manifold is a topological space in which every point has a neighbourhood homeomorphic to Euclidean space $\mathbb{R}^n$. The approach proposed here involves employing an encoder-decoder strategy to acquire actuation manifolds, incorporating ISOMAP (Tenenbaum *et al.*, 2000), neural networks and k-nearest neighbour (kNN) algorithms (Fix & Hodges, 1989). The collected dataset undergoes ISOMAP to discern a low-dimensional embedding. Then, the flow reconstruction procedure involves mapping actuation and sensor information to the manifold coordinates, followed by interpolation among the k-NNs to estimate the corresponding snapshot. An overview of this procedure is illustrated in Figure 1 and detailed in the following.

If we consider our flow fields as vector functions $\boldsymbol{u}(\boldsymbol{x}) = (u(x, y), v(x, y))$ belonging to a Hilbert space, the inner product between two snapshots $\boldsymbol{u}_i$ and $\boldsymbol{u}_j$ is defined by:

$$\langle \boldsymbol{u}_i, \boldsymbol{u}_j \rangle = \iint_\Omega \boldsymbol{u}_i(x, y) \boldsymbol{\cdot} \boldsymbol{u}_j(x, y)\, dx\, dy \tag{1}$$

where "$\boldsymbol{\cdot}$" refers to the scalar product in the two-dimensional vector space. Norms are canonically defined by $\|\boldsymbol{u}\| = \sqrt{\langle \boldsymbol{u}, \boldsymbol{u} \rangle}$ and distances between snapshots are consistent with this norm.

For our study, we have a collection of M flow field snapshots. In the encoder procedure, the ISOMAP algorithm necessitates determining the square matrix $\mathbf{D}_{\mathrm{G}} \in \mathbb{R}^{M \times M}$ containing the geodesic distances among snapshots. Geodesic distances are approximated by selecting a set of neighbours for each snapshot to create a kNN graph. Within these neighbourhoods, snapshots are linked by paths whose lengths correspond to Hilbert-space distances. Geodesic distances between non-neighbours are then obtained as the shortest path through neighbouring snapshots in the kNN graph (Floyd, 1962). At this stage, selecting the number of neighbours, which in the encoding part we denote by k_e, is crucial for creating the kNN graph and thus approximating the geodesic distance matrix.

Existing techniques are mostly empirical as in Samko *et al.* (2006). The method proposed by Samko *et al.* (2006) was successfully used by Farzamnik *et al.* (2023). However, due to the particularity of our dataset, as explained later in the article, standard empirical methods did not yield acceptable results. For this reason, we use the approach based on the Frobenius norm (denoted by $\|\cdot\|_F$) of the geodesic distance matrix $\mathbf{D}_{\mathrm{G}}$ (Shao & Huang, 2005) to determine k_e. Opting for a small k_e can lead to disconnected regions and undefined distances within the dataset. Thus, there exists a minimum value of k_e ensuring that all snapshots are linked in the kNN graph. Starting from this minimum, increasing k_e results in more connections in the kNN graph, causing the Frobenius norm of the geodesic distance matrix to decrease monotonically. However, using an excessively large value may cause a short-circuiting within the kNN graph. Eventually, this yields an Euclidean representation, thus losing ISOMAP's capability to unfold nonlinear relationships within the dataset. Such short-circuiting is indicated by a sudden drop in $\|\mathbf{D}_{\mathbf{G}}\|_F$ with increasing k_e. Thus, the minimum and the value at which short-circuiting first occurs define a suitable range of k_e to be selected to approximate the geodesic distance matrix. The choice of k_e within this range is generally based on how well the high-dimensional data are represented in the low-dimensional space. This is measured in terms of residual variance, as done by Kouropteva *et al.* (2002).

After constructing the geodesic distance matrix $\mathbf{D}_{\mathrm{G}}$, multidimensional scaling (Torgerson, 1952) is employed to construct the low-dimensional embedding $\boldsymbol{\Gamma} = (\tilde{\gamma}_{ij})_{1 \leq i \leq M,\, 1 \leq j \leq n}$, whose dimension is n. The jth column of the matrix $\boldsymbol{\Gamma}$ is the jth ISOMAP coordinate for all the M flow field snapshots. We refer to the ISOMAP coordinates with $\gamma_j\, , j = 1, \ldots, n$. Consequently, $\tilde{\gamma}_{ij} = \gamma_j(i),\ i = 1, \ldots, M$.

More specifically, to obtain the low-dimensional embedding, we need to minimise the cost function:

$$\|\boldsymbol{\Gamma}\boldsymbol{\Gamma}^T - \mathbf{B}\|_F^2 \tag{2}$$

where $\mathbf{B}$ is the double-centred squared-geodesic distance matrix $\mathbf{B} = -1/2\mathbf{H}^T(\mathbf{D}_{\mathrm{G}} \odot \mathbf{D}_{\mathrm{G}})\mathbf{H}$. The centring matrix $\mathbf{H}$ is defined as $\mathbf{H} = \mathbf{I}_M - (1/M)\mathbf{1}_M$. Here $\mathbf{I}_M$ and $\mathbf{1}_M$ are the identity matrix and the matrix with all unit components of size $M \times M$, respectively. The symbol $\odot$ refers to the Hadamard product. The solution of the minimisation term in equation (2) involves the eigenvalues and eigenvectors arising from the decomposition of $\mathbf{B}$, specifically $\mathbf{B} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^T$. For a given n, we select the n largest positive eigenvalues, i.e. $\boldsymbol{\Lambda}_n$, and the corresponding eigenvectors $\mathbf{V}_n$. The coordinates are obtained by scaling these eigenvectors by the square roots of their corresponding eigenvalues. Thus, the coordinates are given by $\boldsymbol{\Gamma} = \mathbf{V}_n\boldsymbol{\Lambda}_n^{1/2}$, where $\boldsymbol{\Lambda}_n^{1/2}$ is a diagonal matrix containing the square roots of the n largest eigenvalues.

In this instance, following the methodology outlined by Tenenbaum *et al.* (2000), the evaluation of data representation quality within the ISOMAP technique is conducted through the residual variance $R_v = 1 - R^2\left(\mathrm{vec}\left(\mathbf{D}_\mathrm{G}\right), \mathrm{vec}\left(\mathbf{D}_\mathrm{E}\right)\right)$. Here $R^2(\cdot,\cdot)$ denotes the squared correlation coefficient, 'vec' is the vectorisation operator and $\mathbf{D}_\mathrm{E}$ represents the matrix of the Euclidean distances between the low-dimensional representation of all the snapshots, retaining only the first n ISOMAP coordinates. The residual variance indicates the ability of the low-dimensional embedding to reproduce the geodesic distances in the high-dimensional space. The dimension n of the low-dimensional embedding is typically determined by identifying an elbow in the residual variance plot.

After the actuation manifold identification, the objective is to perform a flow reconstruction from the knowledge of a reduced number of sensors and actuation parameters. This process is carried out in two steps. First, a regression model is trained to identify the low-dimensional representation of a snapshot. Specifically, we employ a fully connected multi-layer perceptron (MLP) to map the actuation parameters and sensor information to the ISOMAP coordinates. It is possible to employ as network input either the actuation or the Kiki parameters. In the following, we employ the Kiki parameters since we deem them more elegant, although this does not substantially affect the results, being the Kiki parameters a linear combination of the actuation parameters. Second, a decoding procedure is conducted through linear interpolation among a fixed number (denoted by k_d) of nearest neighbours, following the methodology established by Farzamnik *et al.* (2023). An alternative to this mapping is presented in appendix Appendix A, where a two-step kNN regression with distance-weighted averaging is employed. It is important to clarify that the decoding step is applicable only for interpolation cases, meaning for actuation cases that fall within the limits explored during the dataset generation, i.e., $|b_i| < 3,\ \forall i = 1,2,3$, with $|\cdot|$ being the absolute value.

4. Results

4.1. *Identification and physical interpretation of the low-dimensional embedding*

The results of the encoding step are represented in Figure 2. First, the neighbourhood size k_e for the encoding must be determined. The Frobenius norm of the geodesic distance matrix $||\mathbf{D}_\mathrm{G}||_F$ as a function of k_e is reported in Figure 2(a). A minimum $k_e = 40$ is required to ensure that all the neighbourhoods are connected. Figure 2(b) reports the percentage of connected snapshots in the kNN graph. This percentage reaches 100% for $k_e = 40$. Below this value, the manifold is divided into disjointed regions. By increasing k_e starting from the minimum value, the Frobenius norm decreases due to improved connections between neighbourhoods. This also raises the probability of short-circuits, which drastically reduces $||\mathbf{D}_\mathrm{G}||_F$. Indeed, the first short-circuiting is identified for $k_e = 60$. Choosing a value of k_e within the range from the minimum value up to the point where the first significant drop in $||\mathbf{D}_\mathrm{G}||_F$ occurs does not result in substantial alterations of the geometry of the manifold or the physical interpretation of its coordinates. Therefore, in this paper, we select the minimum value, that is $k_e = 40$.

Then the residual variance is used to search for the true dimensionality of the dataset. Employing $n = 5$ coordinates leads to a residual variance of less than 10%, thus $n = 5$ dimensions are deemed sufficient to describe the manifold, as shown in Figure 2(c). Further increase in the number of dimensions provides only marginal changes in the residual variance.

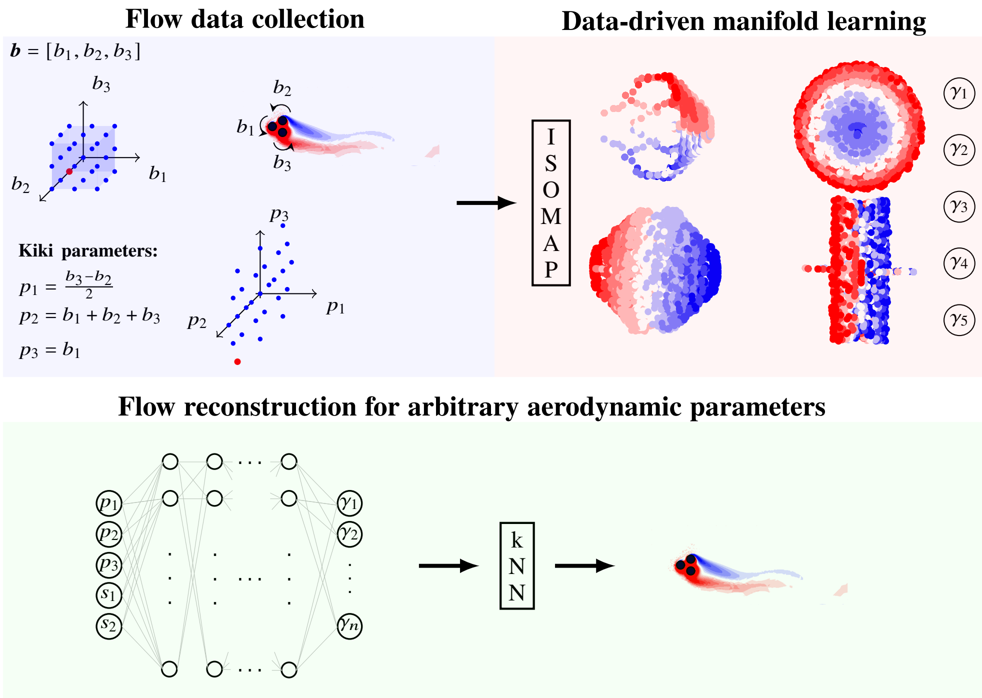

Figure 1: Illustration of the methodology for actuation-manifold learning and full-state estimation. The diagram highlights key steps, from flow data collection to data-driven actuation manifold discovery (upper section). A neural network, incorporating actuation parameters (p_1, p_2 and p_3) and sensor information (s_1 and s_2), determines ISOMAP coordinates $(\gamma_1, \gamma_2, \ldots, \gamma_n)$ and a kNN decoder is used for the full-state flow reconstruction.

The representation of the five-dimensional embedding is undertaken in Figure 2(d) in the form of two-dimensional projections on the $\gamma_i - \gamma_j$ planes with $i, j = 1, ..., 5$ and $i < j$. A first visual observation of the projections highlights interesting physical interpretations. The projection on the $\gamma_3 - \gamma_4$ plane unveils a circular shape, suggesting that these two coordinates describe a periodic feature, i.e. the vortex shedding in the wake of the pinball. The fact that the circle is full suggests that different control actions result in an enhanced or attenuated vortex shedding. Each steady state describes a circle in the $\gamma_3 - \gamma_4$ plane. The radius of each circle explains the amplitude of vortex shedding, while the angular position provides information about the phase. Cases that do not exhibit vortex shedding collapse into points where $\gamma_3, \gamma_4 \approx 0$. This is evident from the observation of the projections on the planes $\gamma_1 - \gamma_3$ and $\gamma_1 - \gamma_4$, both returning a champagne coupe shape, suggesting that smaller values of γ_1 are a prerogative of the cases with limit cycle of smaller amplitude.

This feature also suggests that γ_1 is correlated with the boat tailing parameter p_1 and thus with the drag coefficient C_D, as visualised in Figure 3. This plot also shows a correlation between γ_2, the Magnus parameter p_2 and the lift coefficient C_L. The fifth coordinate of the low-dimensional embedding γ_5, on the other hand, appears to be correlated with the stagnation point parameter p_3 and partially explains the lift produced

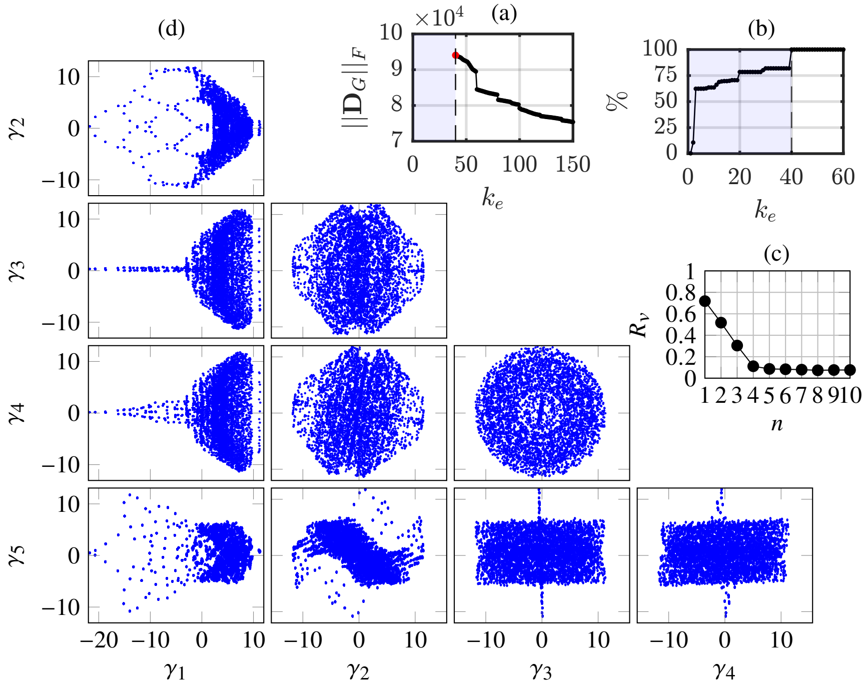

Figure 2: (a) The Frobenius norm of the geodesic distance matrix plotted against the number of neighbours employed in Floyd's algorithm. (b) The ratio between the connected snapshots and the total number of snapshots as k_e increases. Results of the manifold obtained for $k_e = 40$ are presented in panels (c) and (d). The former illustrates the residual variance of the first 10 ISOMAP coordinates, whereas the latter showcases all possible manifold sections identified by the first five coordinates.

by the pinball. Intriguingly, all ISOMAP coordinates are physically meaningful and allow us to discover the three Kiki parameters without human input.

Another interesting physical interpretation arises observing the manifold section $\gamma_1 - \gamma_2$, as shown in Figure 4. This provides insights into the horizontal symmetry of the data. In the figure, semicircles repeat, increasing in number as γ_1 increases. Within each of them, b_1 varies from -3 to 3, whereas b_2 and b_3 remain fixed. The branches symmetrically positioned with respect to the axis $\gamma_2 = 0$ are associated with symmetric actuation b_2 and b_3.

The low-dimensional embedding, being the result of the eigenvector decomposition of the double-centred squared-geodesic distance matrix, is made of orthogonal vectors. While this might represent a disadvantage to autoencoders (Otto & Rowley, 2022), this also allows projecting the snapshots on this basis and obtaining spatial modes that can be used to corroborate the physical interpretation of the manifold coordinates. The jth spatial mode $\boldsymbol{\phi}_j$, $j = 1, \ldots, n$ is a linear combination of the snapshots $\boldsymbol{u}_i$, i.e. $(\sum_{i=1}^{M} \gamma_{ij}^2)^{1/2} \boldsymbol{\phi}_j = \sum_{i=1}^{M} \tilde{\gamma}_{ij} \boldsymbol{u}_i$.

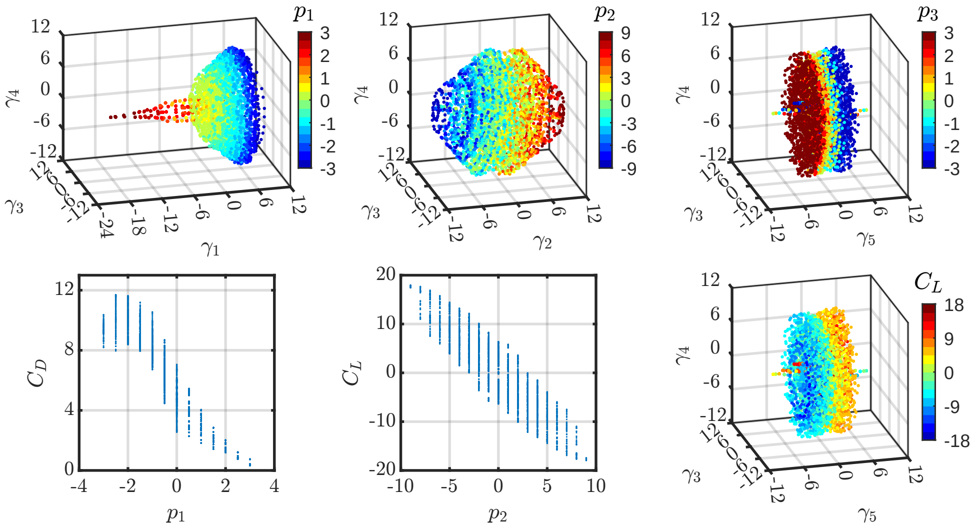

Figure 3: Three-dimensional projections of the manifold colour-coded with actuation parameters and forces coefficients. The first (boat-tailing) and second (Magnus) actuation parameters are plotted against lift and drag coefficients to understand physical control mechanisms.

The first five ISOMAP modes are visualised in Figure 5 with a line integral convolution (LIC; Forssell & Cohen, 1995) plot superimposed on a velocity magnitude contour plot. The visual observation of the modes confirms the interpretation of the physical meaning of the coordinates of the low-dimensional embedding. Three modes, namely $\boldsymbol{\phi}_1$, $\boldsymbol{\phi}_2$ and $\boldsymbol{\phi}_5$ are representative of the actuation parameters. The first ISOMAP mode is characterised by the presence of a jet (wake) downstream of the pinball due to boat-tailing (base bleeding). The second ISOMAP mode represents a circulating motion around the pinball, responsible for positive or negative lift, depending on the circulation direction. The fifth mode is characterised by a net circulation around the front cylinder, determining the position of the front cylinder stagnation point, if added to the mean field. The two spatial modes $\boldsymbol{\phi}_3$ and $\boldsymbol{\phi}_4$, instead, have the classical aspect of vortex shedding modes. Together, they describe the wake response to actuation parameters (far field), providing information on the intensity and phase of vortex shedding.

4.2. *Flow estimation*

For the identification of the low-dimensional representation of a snapshot, we employ a fully connected MLP. The mapping is done having as inputs the Kiki parameters and two sensors. The mapping also includes sensor information because each actuation vector is associated with a limit cycle, which in our case is described by 20 post-transient snapshots. Therefore, the actuation vector provides comprehensive knowledge of the coordinates representing the near field (γ_1, γ_2 and γ_5). Far-field coordinates (γ_3 and γ_4) identification requires information about the phase of vortex shedding, which is provided by the sensors. Two different alternatives are proposed here. In the first, we utilise the lift coefficient

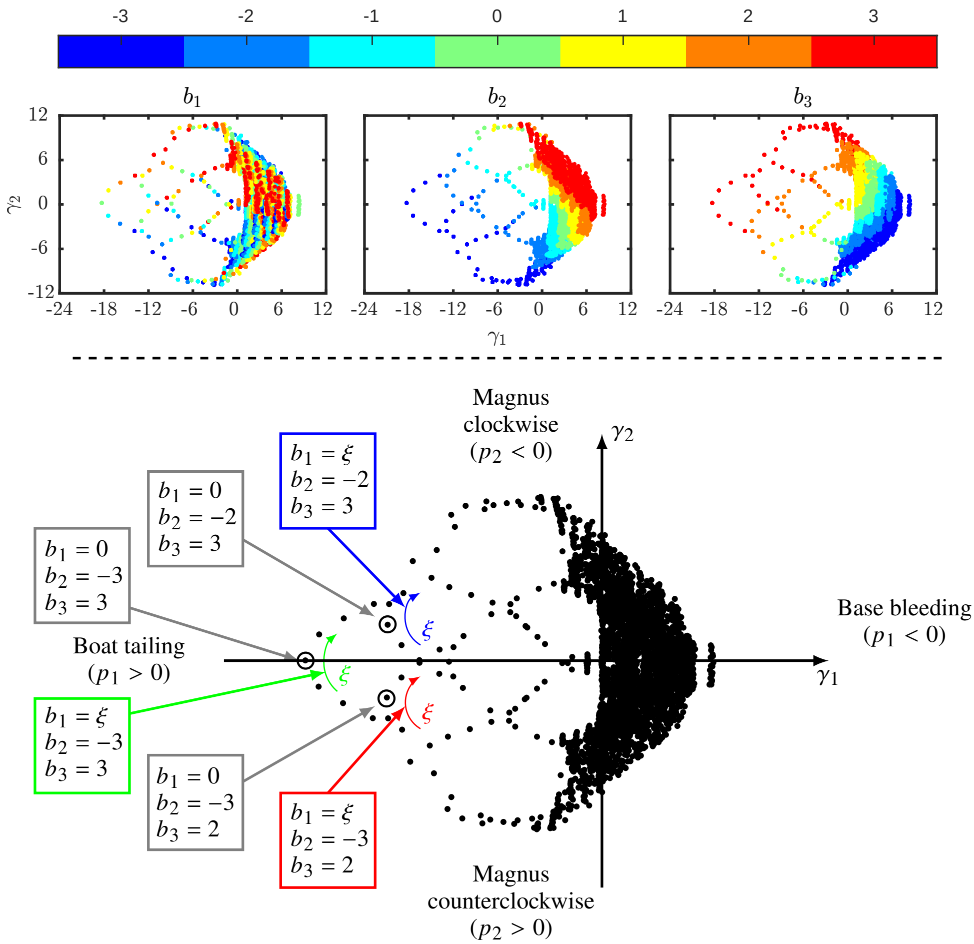

Figure 4: At the top, representation of the manifold section identified by the coordinates γ_1 and γ_2 colour coded with b_1 (left), b_2 (centre) and b_3 (right). At the bottom, a diagram explaining how the coordinate γ_2 provides indications of horizontal symmetry in the flow field.

and its one-quarter mean shedding period delay (we refer to this case with MLP$_1$). In the second, crosswise components of velocity are measured at two positions in the wake, specifically at points $\boldsymbol{x}_1 = (7, 1.25)$ and $\boldsymbol{x}_2 = (10, 1.25)$ (MLP$_2$). The sensors are located in the crosswise position at the edge of the shear layer of the unforced case, i.e. at the top of the upper cylinder. The streamwise location is selected such that the vortex shedding is already developed. The streamwise separation between the two sensors is such that the flight time between them is approximately one-quarter of the shedding period in the unforced case. This ensures that the sensor data collect relevant information for the flow reconstruction, even though a systematic position optimisation has not been performed. The characteristics of the employed networks are summarised in Table 1. The training

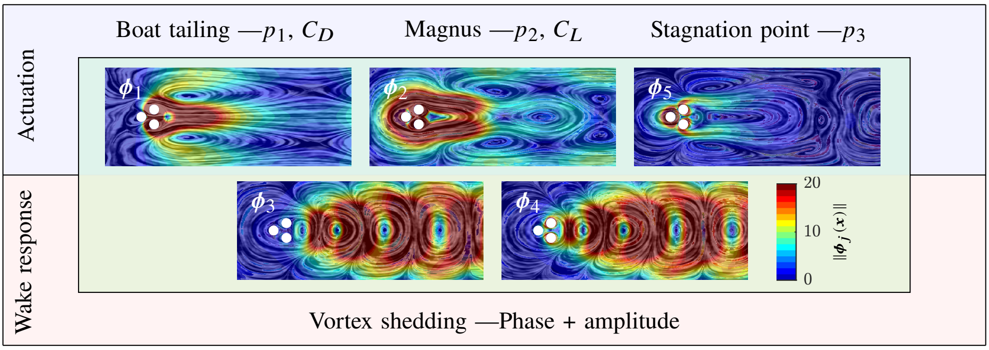

Figure 5: LIC representations of the normalised actuation modes. The shadowed contour represents the local velocity magnitude of the pseudomodes.

Input layer size	5	Output layer size	5
Number of samples	6860	Loss	mean-square-error
Training set	80%	Batch size	2048
Validation set	20%	Number of epochs	Early stop on validation loss
Test set	22 external cases	Optimizer	Adam
I/O scaling	Mean–std $\to [0, 1]$	Activation function	tanh
Hidden layers MLP$_1$	(70,70,70,70)	Hidden layers MLP$_2$	(40,40)

Table 1: Structure of the neural networks used for latent coordinate identification.

of the neural networks is performed with the dataset used to construct the data-driven manifold randomly removing all snapshots related to 20% of the actuation cases and using the remaining snapshots for validation.

To test the accuracy of flow estimation, we use 22 additional simulations with randomly selected actuation, not present in the training nor the validation dataset. As done for the training dataset, the last 20 c.u. are sampled every 1 c.u. for a total of 440 snapshots. The neural networks are used to identify the low-dimensional representation of the snapshots in the test dataset and then the kNN decoder is applied to reconstruct the full state. In the decoding phase, we select $k_d = 220$.

The value of k_d was chosen by performing a parametric study. In particular, we selected the k_d to minimise the reconstruction error of the snapshots in the validation dataset. The selected k_d is considerably higher than the k_e. A reason for this result lies in the particular topology of the present dataset. For some specific control actions, vortex shedding is completely suppressed, resulting in a stationary post-transient solution. The cases with no shedding have 20 snapshots sampled in the post-transient identical to each other. Likewise, their representation in low-dimensional space is identical. When decoding using the kNN interpolation, for snapshots close to those without shedding, considering a low k_d is equivalent to considering a small number of directions for the estimation of the gradients, which is not sufficient and results in a reconstruction with considerable error. To get a practical idea, considering $k_d < 20$ would be equivalent to considering only one

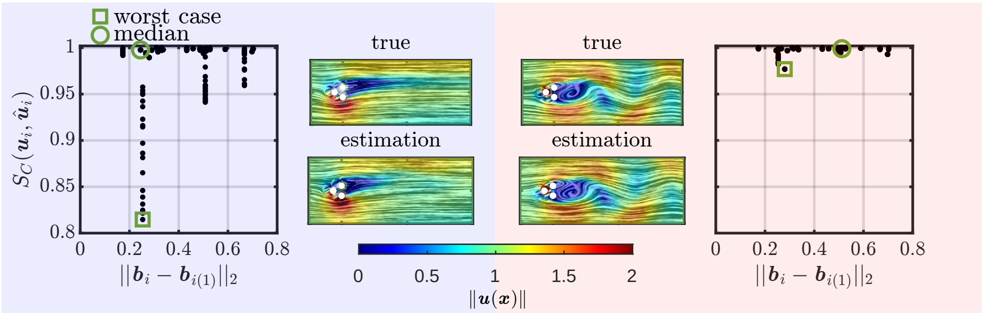

Figure 6: Cosine similarity between real and reconstructed snapshots for all 22 actuation cases in the test dataset. Reconstruction error is plotted against the distance of each actuation case ($\boldsymbol{b}_i$) to its nearest neighbour in the training dataset $\boldsymbol{b}_{i(1)}$. In the center, a comparison between the true snapshot and its estimation is presented for the median error case. Figure presented with LIC and color-coded by velocity magnitude. The number of neighbours is fixed to $k_d = 220$. Results are shown for MLP$_1$ (left) and MLP$_2$ (right).

direction, which is insufficient for decoding. Such a dataset topology greatly complicates the choice of k_d, suggesting the need to identify a k_d that is locally optimal. Such a choice, however, goes beyond the scope of the present research and is deferred to future work.

The accuracy of the estimation between the true ($\boldsymbol{u}_i$) and estimated ($\hat{\boldsymbol{u}}_i$) snapshots is quantified in terms of cosine similarity $S_G(\boldsymbol{u}_i, \hat{\boldsymbol{u}}_i) = \langle \boldsymbol{u}_i, \hat{\boldsymbol{u}}_i \rangle / (\|\boldsymbol{u}_i\| \|\hat{\boldsymbol{u}}_i\|)$. Results can be observed in Figure 6, the knowledge of the manifold enables a full state estimation with few sensors and minimal reconstruction error. When performing the flow reconstruction using MLP$_1$, the median cosine similarity is 0.9977, corresponding to a median root-mean-square error (RMSE) of 0.0512. Using MLP$_2$, instead, the median cosine similarity is 0.9990 (RMSE = 0.0336).

5. Conclusions

In this study, we have addressed a challenge of reduced-order modelling: accurate low-dimensional representations for a large range of operating conditions. The flow data are compressed in a manifold using ISOMAP as encoder and kNN as decoder. The methodology is demonstrated for the fluidic pinball as an established benchmark problem of modelling and control. The cylinder rotations are used as three independent steady control parameters. The starting point of the reduced-order modelling is a data set with 20 statistically representative post-transient snapshots at $Re = 30$ for 343 sets of cylinder rotations covering uniformly a box with circumferential velocities between -3 and 3.

The ISOMAP manifold describes all snapshots with 5 latent variables and few per cent representation errors. To the best of the authors' knowledge, this is the first demonstration of data-driven manifold learning for a flow under multiple input control. Intriguingly, all latent variables are aligned with clear physical meanings. Three coordinates correspond to near-field actuation effects. More precisely, these coordinates are strongly aligned with the Kiki parameters $\boldsymbol{p}$ introduced by Lin (2021), describing (i) the level of boat-tailing (base bleeding) p_1 leading to drag reduction (increase), (ii) the strength of the Magnus effect p_2 leading to a steady lift and characterising (iii) the forward stagnation point p_3.

Two further coordinates correspond to the wake response to actuation, i.e. the amplitude and phase of vortex shedding. The distillation of physically meaningful parameters from a fully automated manifold is surprising and inspiring (see Figure 1).

Our low-rank model is a key enabler for flow estimation with a minimum number of sensors. It is possible to estimate the full flow state with small reconstruction errors just by knowing the actuation parameters and employing two additional measurements, namely the aerodynamic forces or the flow velocity in two points of the wake.

We emphasise that the low-dimensional characterisation of post-transient dynamics as a function of three actuations constitutes a modelling challenge and encourages numerous further applications, e.g. vortex-induced vibration of a cylinder at different spring stiffnesses and cylinder masses, aerodynamic flutter of an elastic airfoil with different wing elasticities, combustion instabilities for different reaction parameters, just to name a few. A low-dimensional manifold representation can be utilised for understanding, estimation, prediction and model-based control. In summary, the presented results can be expected to inspire a large range of applications and complementary manifold learning methods.

Appendix A. Comparison between neural network and kNN performances in the decoding step

In this appendix, we propose an alternative to the decoding as illustrated in Figure 1. In this approach, the decoding is carried out only utilising a kNN regression with weighted averaging with the inverse of distances between neighbours. We transition from actuation parameters and sensor data to the low-dimensional representation of a snapshot using an initial kNN regression, with a number of neighbours indicated as k_{d_1}. Subsequently, we move from the manifold coordinates to the full state reconstruction using another kNN regression, with a number of neighbours indicated as k_{d_2}.

Consider the estimation of a snapshot $\boldsymbol{u}_i$ based on the actuation vector and sensor information in the vector $\boldsymbol{z}_i = (p_1, p_2, p_3, s_1, s_2)^T$. We estimate the low-dimensional representation of this snapshot, which we denoted as $\hat{\boldsymbol{y}}_i$, utilising the first kNN regression. Therefore, starting from $\boldsymbol{z}_i$, we consider its k_{d_1}-nearest vectors in the dataset, denoted as $\boldsymbol{z}_{i(1)}, \dots, \boldsymbol{z}_{i(k_{d_1})}$, and their counterparts in the low-dimensional embedding $\boldsymbol{y}_{i(1)}, \dots, \boldsymbol{y}_{i(k_{d_1})}$ to derive $\hat{\boldsymbol{y}}_i$ as follows:

$$\hat{\boldsymbol{y}}_i = \frac{\sum_{j=1}^{k_{d_1}} \boldsymbol{y}_{i(j)} ||\boldsymbol{z}_{i(j)} - \boldsymbol{z}_i||^{-1}}{\sum_{j=1}^{k_{d_1}} ||\boldsymbol{z}_{i(j)} - \boldsymbol{z}_i||^{-1}} \tag{3}$$

Subsequently, the same procedure is applied to estimate the snapshot, denoted as $\hat{\boldsymbol{u}}_i$. We consider the k_{d_2} nearest vectors to $\hat{\boldsymbol{y}}_i$, labelled as $\hat{\boldsymbol{y}}_{i(1)}, \dots, \hat{\boldsymbol{y}}_{i(k_{d_2})}$, and their counterparts in the high-dimensional space $\boldsymbol{u}_{i(1)}, \dots, \boldsymbol{u}_{i(k_{d_2})}$ to obtain the estimate as follows:

$$\hat{\boldsymbol{u}}_i = \frac{\sum_{j=1}^{k_{d_2}} \boldsymbol{u}_{i(j)} ||\hat{\boldsymbol{y}}_{i(j)} - \hat{\boldsymbol{y}}_i||^{-1}}{\sum_{j=1}^{k_{d_2}} ||\hat{\boldsymbol{y}}_{i(j)} - \hat{\boldsymbol{y}}_i||^{-1}} \tag{4}$$

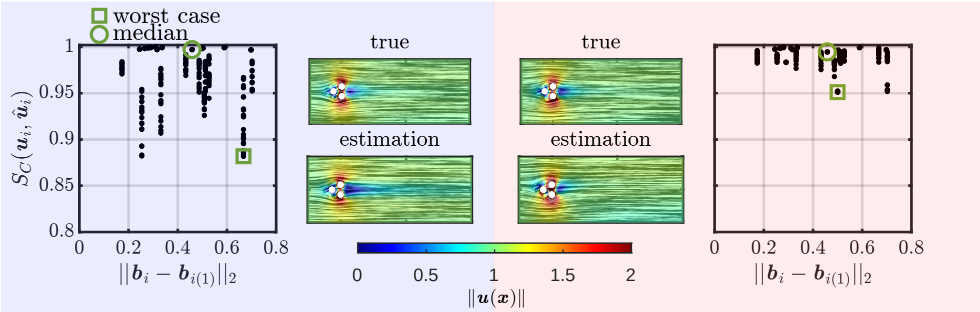

Figure 7: Cosine similarity between real and reconstructed snapshots for all 22 actuation cases in the test dataset. Reconstruction error is plotted against the distance of each actuation case to its nearest neighbour in the training dataset. In the centre, a comparison between the true snapshot and its estimation is presented for the median error case. Figure presented with LIC and color-coded by velocity magnitude. Results are shown force for sensors (left) and velocity sensors (right).

For the selection of k_{d_1} and k_{d_2} we have to solve the minimisation problems in (5):

$$\begin{cases} \underset{k_{d_1}}{argmin} \sum_{i=1}^{M} \left(\hat{\boldsymbol{y}}_i^{(-m),k_{d_1}} - \boldsymbol{y}_i \right)^2 \\ \underset{k_{d_2}}{argmin} \sum_{i=1}^{M} \left(\hat{\boldsymbol{u}}_i^{(-m),k_{d_2}} - \boldsymbol{u}_i \right)^2 \end{cases} \tag{5}$$

where $\hat{\boldsymbol{y}}_i^{(-m),k_{d_1}}$ and $\hat{\boldsymbol{u}}_i^{(-m),k_{d_2}}$ are the estimations of $\boldsymbol{y}_i$ and $\boldsymbol{u}_i$ using the kNN regression with k_{d_1} and k_{d_2} neighbours, respectively, and using as the training dataset all snapshots except the $m = 20$ snapshots of the limit cycle to which the ith snapshot, being estimated, belongs. The procedure follows the same principles of the leave-one-out cross-validation (Wand & Jones, 1994), but instead of eliminating only one case for the estimation, all snapshots of the limit cycle corresponding to the execution are eliminated. The results of the snapshot reconstruction using kNN regression in the two steps of the decoding process are presented in Figure 7.

Table 2 shows the values of k_{d_1} and k_{d_2} used for reconstructing the snapshots in the testing dataset for both sensing cases. In addition, the table also includes the reconstruction errors in terms of median cosine similarity and RMSE. The reconstruction performance is only slightly worse if compared to the case where a neural network followed by kNN interpolation is used.

References

Berkooz, G., Holmes, P. & Lumley, J. L. 1993 The proper orthogonal decomposition in the analysis of turbulent flows. *Annu. Rev. Fluid Mech.* **25** (1), 539–575.

Cornejo Maceda, G. Y., Li, Y., Lusseyran, F., Morzyński, M. & Noack, B. R 2021 Stabilization of the fluidic pinball with gradient-enriched machine learning control. *J. Fluid Mech.* **917**, A42.

			kNN+kNN regression		MLP+kNN interpolation	
	k_{d_1}	k_{d_2}	median S_C	median RMSE	median S_C	median RMSE
Sensing forces	69	7	0.9970	0.0573	0.9977	0.0512
Sensing velocities	66	7	0.9946	0.0804	0.9990	0.0336

Table 2: Comparison of reconstruction errors in terms of cosine similarity and RMSE for both sensing cases using the two decoding methodologies: two-step kNN regression (left) and neural network and kNN interpolation (right). The table also includes the values of k_{d_1} and k_{d_2} used in the two-step kNN regression decoding method.

Deng, N., Noack, B. R., Morzyński, M. & Pastur, L. R. 2020 Low-order model for successive bifurcations of the fluidic pinball. *J. Fluid Mech.* **884**, A37.

Farzamnik, E., Ianiro, A., Discetti, S., Deng, N., Oberleithner, K., Noack, B. R. & Guerrero, V. 2023 From snapshots to manifolds – a tale of shear flows. *J. Fluid Mech.* **955**, A34.

Fernex, D., Noack, B. R. & Semaan, R. 2021 Cluster-based network modeling—From snapshots to complex dynamical systems. *Sci. Adv.* **7** (25), eabf5006.

Fix, E. & Hodges, J. L. 1989 Discriminatory analysis. nonparametric discrimination: Consistency properties. *Int. Stat. Rev.* **57** (3), 238–247.

Floyd, R. W. 1962 Algorithm 97: Shortest path. *Commun. ACM* **5** (6), 345.

Forssell, L. K. & Cohen, S. D. 1995 Using line integral convolution for flow visualization: Curvilinear grids, variable-speed animation, and unsteady flows. *IEEE T. Vis. Comput. Gr.* **1** (2), 133–141.

Haller, G., Kaszás, B., Liu, A. & Axås, J. 2023 Nonlinear model reduction to fractional and mixed-mode spectral submanifolds. *Chaos* **33** (6), 063138.

Helmholtz, H. 1858 Über integrale der hydrodynamischen gleichungen, welche den wirbelbewegungen entsprechen. *J. Reine Angew. Math.* **55**, 25–55.

Holmes, P., Lumley, J. L., Berkooz, G. & Rowley, C. W. 2012 *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*. Cambridge University Press.

Ishar, R., Kaiser, E., Morzyński, M., Fernex, D., Semaan, R., Albers, M., Meysonnat, P. S., Schröder, W. & Noack, B. R. 2019 Metric for attractor overlap. *J. Fluid Mech.* **874**, 720–755.

Kaiser, E., Noack, B. R., Cordier, L., Spohn, A., Segond, M., Abel, M., Daviller, G., Östh, J., Krajnović, S. & Niven, R. K. 2014 Cluster-based reduced-order modelling of a mixing layer. *J. Fluid Mech.* **754**, 365–414.

von Kármán, T. 1912 Über den Mechanismus des Wiederstandes, den ein bewegter Körper in einer Flüssigkeit erfahrt. *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse* **1912**, 547–556.

Kouropteva, O., Okun, O. & Pietikäinen, M. 2002 Selection of the optimal parameter value for the locally linear embedding algorithm. In *FSKD'02, Proceedings of the 1st International Conference on Fuzzy Systems and Knowledge Discovery: Computational Intelligence for the E-Age, Singapore* (ed. L. Wang, S. K. Halgamuge & X. Yao), pp. 359–363.

Li, Y., Cui, W., Jia, Q., Li, Q., Yang, Z., Morzyński, M. & Noack, B. R. 2022 Explorative gradient method for active drag reduction of the fluidic pinball and slanted ahmed body. *J. Fluid Mech.* **932**, A7:1–48.

Lin, Q. 2021 面向控制的全自动降阶模型–以流体弹球为例 (Eng: Fully automated control-oriented reduced-order modeling exemplified for the fluidic pinball). Master's thesis, Harbin Institute of Technology.

Loiseau, J.-Ch., Noack, B. R. & Brunton, S. L. 2018 Sparse reduced-order modeling: Sensor-based dynamics to full-state estimation. *J. Fluid Mech.* **844**, 459–490.

Marusic, I. & Broomhall, S. 2021 Leonardo da Vinci and fluid mechanics. *Annu. Rev. Fluid Mech.* **53**, 1–25.

Noack, B. R., Afanasiev, K., Morzyński, M., Tadmor, G. & Thiele, F. 2003 A hierarchy of low-dimensional models for the transient and post-transient cylinder wake. *J. Fluid Mech.* **497**, 335–363.

Noack, B. R., Ehlert, A., Nayeri, C. N. & Morzynski, M. 2023 Analysis, modeling, and control of the cylinder wake. In *Data-Driven Fluid Mechanics: Combining First Principles and Machine Learning* (ed. M. A. Mendez, A. Ianiro, B. R. Noack & S. L. Brunton), pp. 3–19. Cambridge University Press.

Otto, S. E. & Rowley, C. W. 2022 Inadequacy of linear methods for minimal sensor placement and feature selection in nonlinear systems: A new approach using secants. *J. Nonlinear Sci.* **32** (5), 69.

Protas, B. 2004 Linear feedback stabilization of laminar vortex shedding based on a point vortex model. *Phys. Fluids* **16** (12), 4473–4488.

Raibaudo, C., Zhong, P., Noack, B. R. & Martinuzzi, R. J. 2020 Machine learning strategies applied to the control of a fluidic pinball. *Phys. Fluids* **32**, 015108:1–13.

Samko, O., Marshall, A.D. & Rosin, P.L. 2006 Selection of the optimal parameter value for the isomap algorithm. *Pattern Recognit. Lett.* **27** (9), 968–979.

Shao, C. & Huang, H. 2005 Selection of the optimal parameter value for the isomap algorithm. In *MICAI 2005: Advances in Artificial Intelligence* (ed. Alexander Gelbukh, Álvaro de Albornoz & Hugo Terashima-Marín), pp. 396–404. Springer Berlin Heidelberg.

Stuart, J. T. 1958 On the non-linear mechanics of hydrodynamic stability. *J. Fluid Mech.* **4** (1), 1–21.

Tenenbaum, J. B., de Silva, V. & Langford, J. C. 2000 A global geometric framework for nonlinear dimensionality reduction. *Science* **290** (5500), 2319–2323.

Theofilis, V. 2011 Global linear instability. *Annu. Rev. Fluid Mech.* **43**, 319–352.

Torgerson, W. S. 1952 Multidimensional scaling: I. Theory and method. *Psychometrika* **17** (4), 401–419.

Wand, M. P. & Jones, M. C. 1994 *Kernel Smoothing*. *Chapman & Hall/CRC Monographs on Statistics & Applied Probability* 60. Boca Raton, FL, U.S.: Chapman & Hall.

Wang, X., Deng, N., Cornejo Maceda, G. Y. & Noack, B. R 2023 Cluster-based control for net drag reduction of the fluidic pinball. *Phys. Fluids* **35** (2), 023601.