VILNIUS GEDIMINAS TECHNICAL UNIVERSITY

FACULTY OF MECHANICS

DEPARTMENT OF MECHATRONICS, ROBOTICS AND DIGITAL MANUFACTURING

APPROVED BY
Head of Department

____
(Signature)
Vytautas Bučinskas
(Name, Surname)

____
(Date)

Ahmed Hammouda

CONTROL OF THE FLUIDIC PINBALL THROUGH GENETIC PROGRAMMING

Final Master's thesis

Study programme MECHATRONIC SYSTEMS,
Code 6211EX053
Fields of study PRODUCTION ENGINEERING

Vilnius, 2025

VILNIUS GEDIMINAS TECHNICAL UNIVERSITY

FACULTY OF MECHANICS

DEPARTMENT OF MECHATRONICS, ROBOTICS AND DIGITAL MANUFACTURING

APPROVED BY
Head of Department

____
(Signature)
Vytautas Bučinskas
(Name, Surname)
____
(Date)

Ahmed Hammouda

CONTROL OF THE FLUIDIC PINBALL THROUGH GENETIC PROGRAMMING

Final Master's thesis

Study programme MECHATRONIC SYSTEMS,
Code 6211EX053
Fields of study PRODUCTION ENGINEERING

Supervisor Prof. Darius Virzonis ______ _______
(Title, Name, Surname) (Signature) (Date)

Consultant ____________________ __________ ______
(Title, Name, Surname) (Signature) (Date)

Vilnius, 2025

VILNIUS GEDIMINAS TECHNICAL UNIVERSITY

Ahmed Ashraf Badreldin Hammouda, 20230174

(Student's given name, family name, certificate number)

Faculty of Mechanics

(Faculty)

Mechatronics Systems, MSfmu-23

(Study programme, academic group no.)

DECLARATION OF AUTHORSHIP IN THE FINAL DEGREE PROJECT

May 15, 2025

I declare that my Final Degree Project entitled „Control of the Fluidic Pinball through Genetic Programming" is entirely my own work. I have clearly signalled the presence of quoted or paraphrased material and referenced all sources.

I have acknowledged appropriately any assistance I have received by the following professionals/advisers: Prof Doctor Darius Viržonis.

The academic supervisor of my Final Degree Project is Prof Doctor Darius Viržonis.

No contribution of any other person was obtained, nor did I buy my Final Degree Project.

(Signature)	Ahmed Ashraf Badreldin Hammouda (Given name, family name)

VILNIUS GEDIMINAS TECHNICAL UNIVERSITY
FACULTY OF MECHANICS
DEPARTMENT OF MECHATRONICS, ROBOTICS AND DIGITAL MANUFACTURING

Production and Manufacturing Engineering study field
Mechatronics Systems study programme, state code 6211EX053
………………………………………………………. specialisation

APPROVED BY
Head of Department
Vytautas Bučinskas
2025-06-02

OBJECTIVES FOR MASTER THESIS

No. MSfmu-23–9667

Vilnius

For student Ahmed Ashraf Badreldin Hammouda

Master Thesis title: Control of the Fluidic Pinball through Genetic Programming

Deadline for completion of the final work according to the planned study schedule.

THE OBJECTIVES:

Data: The goal is to develop a framework using genetic programming to evolve controllers for nonlinear dynamical systems, focusing on the fluidic pinball problem. The fluidic pinball is a complex fluid dynamics problem involving a set of three cylinders arranged in an equilateral triangle, creating a dynamic and chaotic flow regime around them. This challenging setup requires adaptive control due to its sensitivity to initial conditions and environmental perturbations, making it an ideal testbed for genetic programming-based controllers.

Explanatory Note:
The simulation will be executed on the university’s simulation computer, utilizing a virtual machine to host the genetic programming toolkit and the fluidic pinball simulator. Initially, a range of 100 to 1,000 controllers will be generated, each evolving over 10 to 20 generations with breeding between generations to enhance performance. These ranges allow for adjustment to suit the specific dynamics of each tested configuration.
Controllers will be evaluated against a predefined cost function, which will assess their efficiency in meeting control objectives like minimizing turbulence or stabilizing flow around the three cylindrical obstacles. Based on this cost function, the most efficient controllers will be selected for breeding, ensuring that only the most promising solutions advance to subsequent generations. Simulation parameters such as flow velocity, cylinder spacing, and fluid viscosity are calibrated for realism.

Academic Supervisor Professor Darius Viržonis

Vilniaus Gedimino technikos universitetas	ISBN
Mechanikos fakultetas	Egz. sk. ....
Mechatronikos, robotikos ir skaitmeninės gamybos katedra	Data ..........-.....-.....

ISSN

Antrosios pakopos studijų **Mechatroninių sistemų** programos magistro baigiamasis darbas 4	
Pavadinimas	**Nespūdaus disko skysčio sraute valdymo modeliavimas, naudojant genetinio programavimo algoritmą**
Autorius	**Ahmed Ashraf Badreldin Hammouda**
Vadovas	**Darius Viržonis**

Kalba: anglų

Anotacija

Šiame magistro darbe nagrinėjamas genetinio programavimo (GP) taikymas kuriant pažangias valdymo strategijas netiesiniam ir sudėtingam skysčių dinamikos „fluidic pinball" (skysčių smeigtukų) sistemai, siekiant sumažinti srauto pasipriešinimą (drag) užnugario srityje. Ši sistema naudojama kaip etalonas įvairioms pramoninėms sritims, tokioms kaip mikrofluidika, cheminiai reaktoriai ir automobilių technologijos. Skirtingai nei valdymo metodai, kuriems reikia sudėtingo modeliavimo ir tiesinimo, dažnai įnešančių netikslumų, GP leidžia tiesiogiai evoliucionuoti valdymo dėsnius, kurie prisitaikydami stabilizuoja srautą, remdamiesi slėgio ir greičio jutiklių duomenimis. Darbe išsamiai aprašomas GP procesas ir jo našumo palyginimas su tradiciniais valdymo metodais. Simuliacijų rezultatai rodo, kad GP pagrindu sukurti valdymo dėsniai lenkia optimalius linijinius valdiklius tiek pasipriešinimo mažinimo, tiek energijos efektyvumo požiūriu, siūlydami perspektyvią kryptį netiesiniam srautų valdymui.

Prasminiai žodžiai: genetinis programavimas, skysčių smeigtukai (fluidic pinball), netiesinis valdymas, srauto stabilizavimas, pasipriešinimo mažinimas, mašininio mokymosi valdymas

Vilnius Gediminas Technical University	ISBN / ISSN
Faculty of Mechanics	Copies No. ....
Department of Mechatronics, Robotics and Digital Manufacturing	Date .........-.....-.....

Master Degree Studies **Mechatronics Systems** study programme Master Graduation Thesis 4	
Title	**Control of the Fluidic Pinball through Genetic Programming**
Author	**Ahmed Ashraf Badreldin Hammouda**
Academic supervisor	**Darius Viržonis**

Thesis language: English

Annotation

This thesis explores the use of genetic programming (GP) to develop advanced control strategies for the nonlinear and complex fluidic pinball system, with the goal of reducing drag in the wake flow. The system serves as a benchmark for various industrial applications such as microfluidics, chemical reactors, and automotive systems. Unlike control methods that require complex modeling and linearization, which often introduce inaccuracies, GP directly evolves control laws that adaptively stabilize the flow through pressure and velocity sensor readings. The study details the GP process and compares its performance against conventional control methods. Simulation results show that GP-based control laws outperform optimal linear control laws in terms of drag reduction and energy efficiency, offering a promising direction for nonlinear flow control.

Keywords: genetic programming, fluidic pinball, nonlinear control, flow stabilization, drag reduction, machine learning control

TABLE OF CONTENT

LIST OF FIGURES

NOMENCLATURE

Dimensions		**Units**
$\boldsymbol{N_i}$	Total number of individuals in a generation	**Count**
$\boldsymbol{N_p}$	Tournament size	**Count**
$\boldsymbol{N_{gp}}$	Number of GP generations	**Count**
Indices		
$\boldsymbol{i_a}$	Index for state vector, $\mathrm{i_a} = 1, \dots, \mathrm{n_a}$	**Count**
$\boldsymbol{i_b}$	Index for actuation vector, $\mathrm{i_b} = 1, \dots, \mathrm{n_b}$	**Count**
$\boldsymbol{i_s}$	Index for sensor vector, $\mathrm{i_s} = 1, \dots, \mathrm{n_s}$	**Count**
$\boldsymbol{i_{rnk}}$	Rank of individual in a population, $\mathrm{i_{rnk}} = 1, \dots, N_i$	**Count**
$\boldsymbol{i_{gen}}$	Index of generation, $\mathrm{i_{gen}} = 1, \dots, N_{gp}$	**Count**
$\boldsymbol{i_{run}}$	Index of test run of GP algorithms, $\mathrm{i_{run}} = 1, \dots, N_{\mathrm{run}}$	**Count**
Scalars and constants		
$\boldsymbol{C_d}$	Drag coefficient	**-**
$\boldsymbol{C_0}$	Base drag coefficient	**-**
$\boldsymbol{k}$	Scaling constant for the velocity field difference	**-**
$\boldsymbol{\rho}$	Fluid density	$\boldsymbol{\frac{kg}{m^3}}$
$\boldsymbol{A}$	Reference area of object perpendicular to fluid flow direction	$\boldsymbol{m^2}$
$\boldsymbol{P_r}$	Replication probability	**-**
$\boldsymbol{P_m}$	Mutation probability	**-**
$\boldsymbol{P_c}$	Crossover probability	**-**
$\boldsymbol{\gamma}$	Actuation cost penalization coefficient	**-**
$\boldsymbol{E}$	Energy	$\boldsymbol{J}$
$\boldsymbol{P_a}$	Actuated power	$\boldsymbol{W}$

$\boldsymbol{P_d}$	Drag Power	$\boldsymbol{W}$
Vectors		
$\boldsymbol{F_d}$	Drag force	$\boldsymbol{N}$
$\boldsymbol{u}$	Fluid velocity	$\boldsymbol{\frac{m}{s}}$
$\boldsymbol{a}$	State vector, $a \in \mathbb{R}^{n_a}$	$\boldsymbol{\frac{m}{s}}$
$\boldsymbol{b}$	Actuation vector, $b \in \mathbb{R}^{n_b}$	$\boldsymbol{\frac{m}{s}}$
$\boldsymbol{s}$	Sensor vector, $s \in \mathbb{R}^{n_s}$	$\boldsymbol{\frac{m}{s}}$
$\boldsymbol{T}$	Time vector, $T \in \mathbb{R}^{n_t}$	$\boldsymbol{s}$
$\boldsymbol{u_b}$	Actuated velocity field	$\boldsymbol{\frac{m}{s}}$
$\boldsymbol{u_s}$	Steady solution velocity field	$\boldsymbol{\frac{m}{s}}$
$\boldsymbol{F_{i_s}^{\theta}}$	Azimuth angle component of the fluid's local forces on the pinball cylinders	$\boldsymbol{rad}$
$\boldsymbol{U_x}$	Flow velocity component in drag direction	$\boldsymbol{\frac{m}{s}}$
$\boldsymbol{F_{p,i}}$	Pressure force contribution on the surface of cylinder $\boldsymbol{i}$	$\boldsymbol{N}$
$\boldsymbol{F_{v,i}}$	Viscous force contribution on the surface of cylinder $\boldsymbol{i}$	$\boldsymbol{N}$
Functions		
$\boldsymbol{J}$	Cost function	-
$\boldsymbol{J_a}$	Averaged state cost function	-
$\boldsymbol{J_b}$	Averaged actuation cost function	-
$\boldsymbol{F}$	Plant dynamics function	-
$\boldsymbol{H}$	Measurement function	-
$\boldsymbol{K}$	Arbitrary control law function	-
$\boldsymbol{K^*}$	Optimal control law function	-

$\boldsymbol{K}_{\boldsymbol{i_{run}}}^{\boldsymbol{i_{gen}},\boldsymbol{i_{rnk}}}$	MLC individual of rank $\boldsymbol{i_{rnk}}$ at generation $\boldsymbol{i_{gen}}$ of run $\boldsymbol{i_{run}}$	**-**

INTRODUCTION

Relevance of the topic:

Controlling nonlinear fluid dynamical systems is a key challenge in engineering, impacting fields like aerospace, automotive, and renewable energy. In these areas, fluid flow regulation can significantly affect efficiency and operational costs. For example, vortex shedding, where rotating air vortices form behind objects, increases drag and reduces stability, which is particularly problematic for vehicles and aircrafts, leading to higher fuel consumption and hundreds of millions of dollars in costs each year [1,2,3]. Beyond transportation, fluid dynamics influences manufacturing and energy production. Unsteady fluid flows can disrupt industrial processes. In microfluidic systems, such as lab-on-a-chip devices, controlling fluid flow is vital for precise chemical reactions and biomedical applications [4].

Problem:

Vortex shedding behind automobiles moving through fluid systems results in increased drag force that acts opposite to the movement of the vehicle. The leads to significant energy loss and increased fuel costs.

Research Object:

The fluidic pinball system, a nonlinear flow control setup consisting of three cylinders used to study wake dynamics and active flow control strategies.

Aim:

This project leverages genetic programming (GP) based machine learning control (MLC) to design and generate nonlinear control laws to stabilize the actuated velocity field for the fluidic pinball system without resorting to linearization or mathematical approximations.

Tasks:

1. **Drag Reduction:** Reduce the drag force by minimizing the drag coefficient using genetic programming (GP) and compare the results to those of a traditional linear control model.
2. **Model-Free Control Development**: Apply genetic programming directly to the nonlinear fluidic pinball system to evolve control laws without requiring an explicit mathematical model.
3. **Nonlinear Feedback Adaptation**: Demonstrate that GP can generate effective controllers without linearizing the system, by adapting directly to feedback from its nonlinear dynamics.

Research Methods:

The study uses GP as the primary method to evolve control laws for the fluidic pinball system. Simulations of the system's nonlinear dynamics are performed to evaluate control-law performance. The GP process includes exploration, evaluation, selection, and genetic modification, iteratively improving controllers based on real-time system feedback.

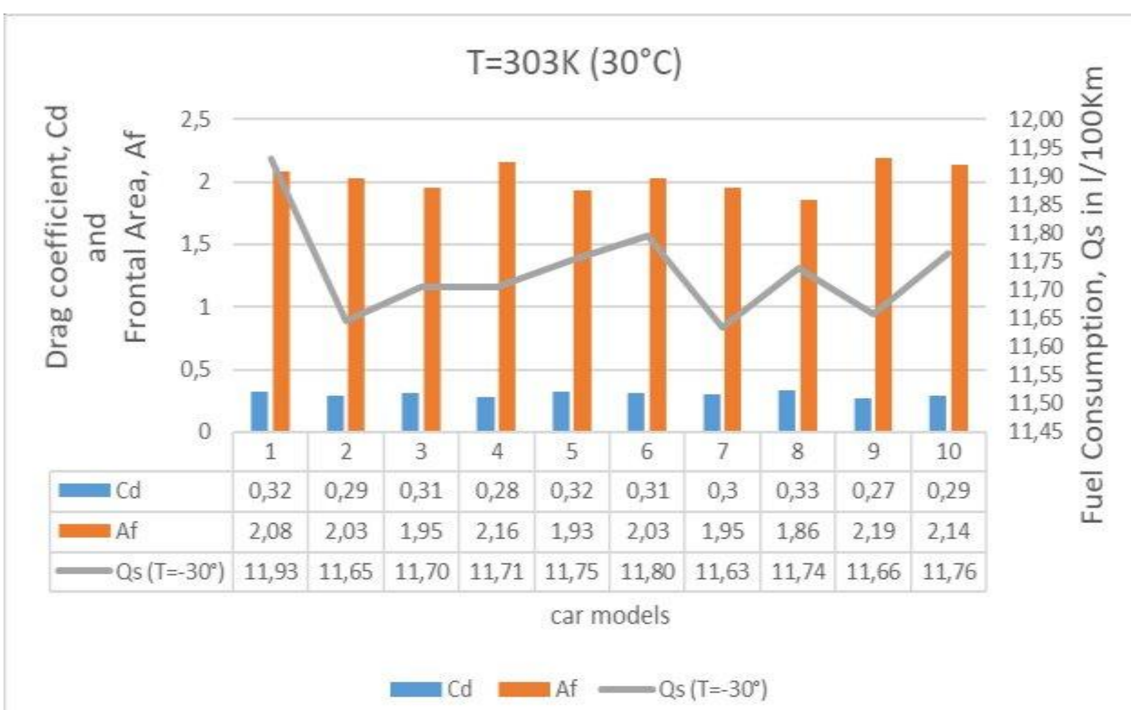
1. LITERATURE REVIEW

This literature review explores the development and application of GP for MLC in fluid systems. It begins by examining early approaches to controlling nonlinear systems and later introduces the previous use of GP on simple mechanical systems.

1.1 Background and Previous Work

1.1.1 Drag Force and Fuel Consumption

Drag force is an external force exerted by a fluid stream on any object moving through a fluid. In fluid dynamics, this drag is amplified by the vortex shedding and wake turbulence behind an automobile. A study by Lawrence Livermore National Laboratory suggests that improving automobile wake dynamics could reduce fuel consumption by 12%, equating to over $10 billion in fuel savings per year [5]. In their study, Ghasemi and Yousefi et al. explore fuel consumption of ground-vehicles under different conditions [6]. Figure 1.1 highlights their critical and major findings in the difference in fuel consumption caused by different automobiles with varying properties in daily automobile transportation. As displayed by the figure and data, a minor decrease in the drag coefficient C_d from $\mathbf{0.32}$ to $\mathbf{0.29}$ (along with a slight frontal reference area A change) can mean a staggering $\mathbf{0.28}\boldsymbol{L/100Km}$ difference between model 1 and 2. One can imagine how this difference is amplified with millions of $\boldsymbol{Km}$ driven daily by people across the globe.



	1	2	3	4	5	6	7	8	9	10
Cd	0,32	0,29	0,31	0,28	0,32	0,31	0,3	0,33	0,27	0,29
Af	2,08	2,03	1,95	2,16	1,93	2,03	1,95	1,86	2,19	2,14
Qs (T=-30°)	11,93	11,65	11,70	11,71	11,75	11,80	11,63	11,74	11,66	11,76

Fig. 1.1. Fuel consumption in automobiles with reference area A and drag coefficient C_D [6]

In calculable parameters, the main cause behind vortex shedding and an increased drag force is due to insufficient tools and control methods to stabilize the actuated velocity field $\boldsymbol{u_b}$ relative to the steady velocity field $\boldsymbol{u_s}$, which greatly increases fuel consumption. Equations 1.1 and 1.2 further explain the relationship between drag and the actuated velocity field in fluid dynamics [7]:

$$\boldsymbol{F_d = \frac{1}{2} C_d \rho A u^2} \tag{1.1}$$

where $\boldsymbol{F_d}$ is the drag force, $\boldsymbol{C_d}$ is the drag coefficient, $\boldsymbol{\rho}$ is the fluid density, $\boldsymbol{A}$ is the reference area of the object perpendicular to fluid flow, and $\boldsymbol{u}$ is the flow velocity. When specifically considering the drag coefficient $\boldsymbol{C_d}$ [8]:

$$\boldsymbol{C_d = C_0 + k\|u_b - u_s\|^2} \tag{1.2}$$

$\boldsymbol{C_0}$ is the base drag coefficient when the velocity fields are perfectly aligned $(\boldsymbol{u_b = u_s})$, and $\boldsymbol{k}$ is a proportional constant that links the deviation between the two velocity fields. Through this relationship, we can surmise that when $\boldsymbol{u_b}$ closely matches $\boldsymbol{u_s}$, the drag coefficient $\boldsymbol{C_d}$ is reduced, thereby decreasing the applied drag force $\boldsymbol{F_d}$**.** Since this is a direct relationship between the two parameters, a significant portion of the losses in transportation systems are due to unstable wake dynamics.

The challenge is that traditional control methods struggle to account for the highly nonlinear nature of these flows. Without an effective control strategy, industries will continue facing escalating operational costs, fuel waste, and performance limitations due to poorly regulated wake dynamics.

1.1.2 Linear Controllers in Fluid Systems

Nonlinear dynamical systems, including fluidic systems, present unique challenges due to their unpredictability, chaotic transitions, and sensitivity to initial conditions. These characteristics complicate traditional control approaches, which often involve linearizing the system equations to simplify the design of linear controllers.

Smith et al. [9] investigated linear controllers for managing wake interactions in a canonical cylinder arrangement. While these controllers performed adequately for small perturbations, they struggled with larger disturbances and unsteady wake patterns. The study concluded that linear control laws are inherently limited in addressing highly nonlinear behaviors, highlighting the need for more adaptive solutions.

With the presented significant control challenge, control methods that rely on linearization or simplified models, such as linear quadratic regulators (LQR), struggle to capture the true nonlinear dynamics of these systems [10]. To further elaborate on this statement, we must first highlight the difference between the linear control law design process vs. the GP-based process. The structure diagram in figure 1.2 demonstrates the difference in the controller-design process of both methods.

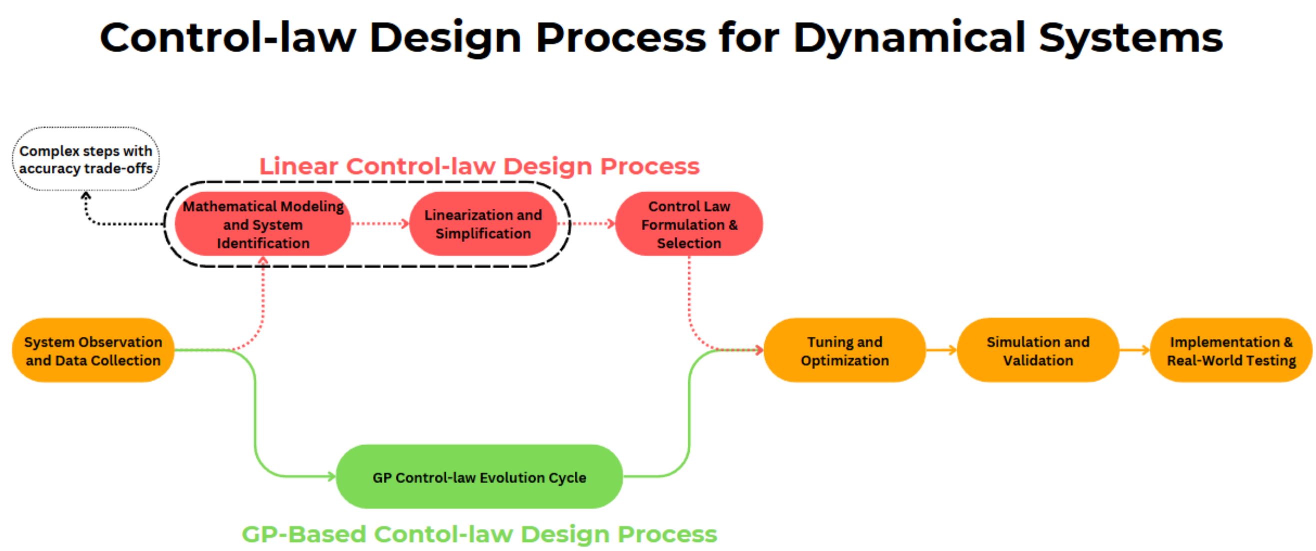


Fig. 1.2. Structure diagram highlighting process difference between GP and linear control methods

As demonstrated in figure 1.2, the main challenge with using linear control methods to find control solutions for nonlinear, dynamical systems lies in two separate steps of the process: **mathematical modeling** and **linearization**.

- **Complexity of mathematical modeling:** This is an important step in the many control law design processes, which includes deriving the equations of motions and developing the state-space model and system transfer function. The modeling stage can prove quite difficult when dealing with dynamical systems that comprise many degrees of freedom.

- **Inaccuracies introduced during linearization:** The next stage after the modeling process is the linearization of the system dynamics. Linear control laws, such as proportional-integral-derivative (PID) controllers, linear quadratic regulators (LQR), and state vector control (also known as K-vector control), can only be practically applied to a linearized model of the system dynamics. However, almost all real-world physical systems have nonlinear parameter relationships, and linearizing/simplifying such dynamics leads to inaccuracies. One can even argue that linear control methods provide control solutions to a "derived similarity" of the dynamical system, and do not represent or duplicate the nonlinear system response.

This results in ineffective or suboptimal control strategies. Data-driven approaches such as machine learning control (MLC) and neural networks offer a promising solution, directly optimizing control laws based on observed behaviors. However, existing methods are insufficient to manage the full complexity of nonlinear fluid systems. The fluidic pinball system serves as a benchmark for studying systems with such complicated dynamics, but traditional techniques often fail to exploit its nonlinear characteristics, underscoring the need for more effective approaches.

1.1.3 Genetic Programming in Mechanical Systems

GP has demonstrated remarkable promise in solving control problems for simple mechanical systems, establishing a foundation for tackling more complex domains.

Case Study 1: Double Pendulum

The double pendulum is an example of a chaotic system, characterized by sensitive dependence on initial conditions and highly nonlinear dynamics. Benchmark studies by White, Smith and Jones et al. utilized GP to develop control strategies for stabilizing the pendulum in its inverted position [11,12]. This method was uniquely named the Improved Genetic Algorithm (IGA). IGA included encoding potential control laws as tree-based structures, and iteratively optimized the control parameters based on a fitness cost function that minimized deviation from the target inverted pendulum position. The fitness used in the study is characterized by cost $\boldsymbol{E(Z)}$:

$$\boldsymbol{min\big(E(Z)\big)}, \qquad \boldsymbol{E(Z) = \sum_{i=1}^{2}\sum_{j=1}^{n}\Big(\theta_i(j) - \hat{\theta}_i(j)\Big)^2} \tag{1.3}$$

where $\boldsymbol{\theta_i(j)}$ is simulated zero-input response data and $\boldsymbol{\widehat{\theta}_i(j)}$ is the real-time zero-input response data of the $\boldsymbol{i}$**th** rod at moment $\boldsymbol{j}$. Therefore, the effectiveness of the control law is characterized by a small cost $\boldsymbol{E(Z)}$, ideally close to zero [11].

In addition, the study observed the difference between the IGA and a linear method of control like the LQR. When considering an LQR controller, it is designed for linear systems or in most cases linearized versions of nonlinear systems, such as the double inverted pendulum. It assumes the system dynamics can be represented by a linear state-space model [13]:

$$\boldsymbol{\dot{x}(t) = Ax(t) + Bu(t)} \tag{1.4}$$

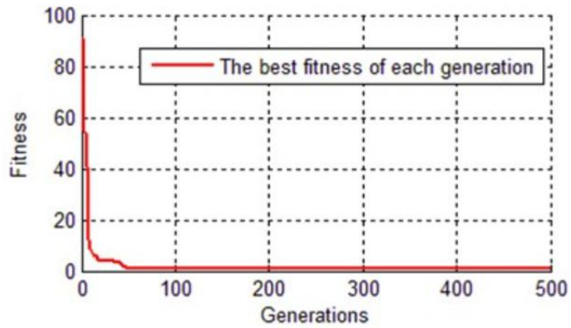
where $\boldsymbol{x(t)}$ is the state vector, $\boldsymbol{u(t)}$ is the control input (which in this inverted pendulum system corresponds to motor voltage $\boldsymbol{V}$), and $\boldsymbol{A}$ and $\boldsymbol{B}$ are the linearized system matrices that correspond to system dynamics [12]. On the other hand, the IGA is applied directly to the nonlinear dynamics of the pendulum:

$$\boldsymbol{\dot{x}(t) = f\big(x(t), u(t)\big)} \tag{1.5}$$

where $\boldsymbol{f\big(x(t), u(t)\big)}$ is the nonlinear function describing the accurate system dynamics.

Key Results:

- The IGA-evolved control law stabilized the double pendulum in over 85% of test cases, even under varying initial conditions. Even though the algorithm ran for 500 generations, the fitness of the optimized control law converged in an impressive 20-generation span, as shown in figure 1.3.



Generation (G)	Fitness $E(Z)$
1	90.802
20	4.0011
40	3.4501
60	1.1053
80	1.0658
100	1.0658
200	1.0057
300	0.9271
400	0.8629
500	0.8515

Fig. 1.3. Convergence of control law fitness over 500 generations on the double pendulum system [13]

- Compared to linear control methods, such as the LQR, the IGA achieved a 30% improvement in angle control of the outer pendulum arm, measured by the angle error difference in radians displayed in figure 1.4.

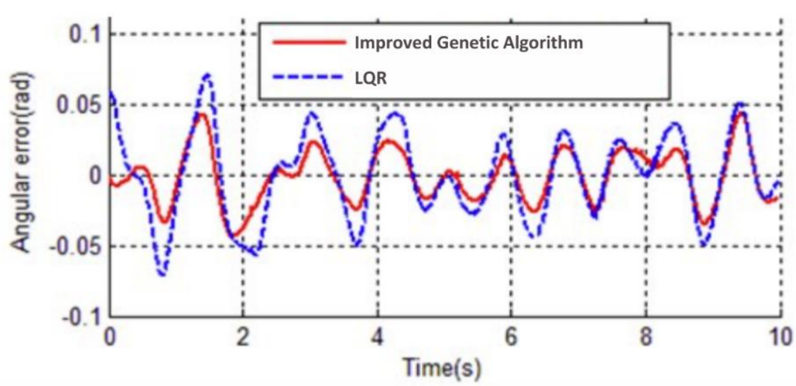


Fig. 1.4. Recorded angular error between the adapted IGA algorithm and LQR [13]

- Computational complexity was significantly reduced, as the IGA approach eliminated the need for explicitly modeling of the system's equations of motion as compared to its LQR alternative.

Case Study 2: Cart-Pole System

The cart-pole system, often referred to as an inverted pendulum on a moving base, is another well-known benchmark for testing control strategies. In a recent study, GP was employed to evolve control laws capable of stabilizing the pole while maintaining the cart's position within predefined bounds [14]. However, the most important observed outcome from this study was not the robustness of the control law's performance when maintaining the inverted pendulum angle $\boldsymbol{\theta}$, but in the movement pattern necessary for initially inverting the pendulum to its starting position.

Key Results:

- After running over 400 generations worth of individuals, the evolved control law's robustness and performance were displayed in the optimal path for the initial pendulum swerve to the inverted initial position. This path is displayed in figure 1.5.

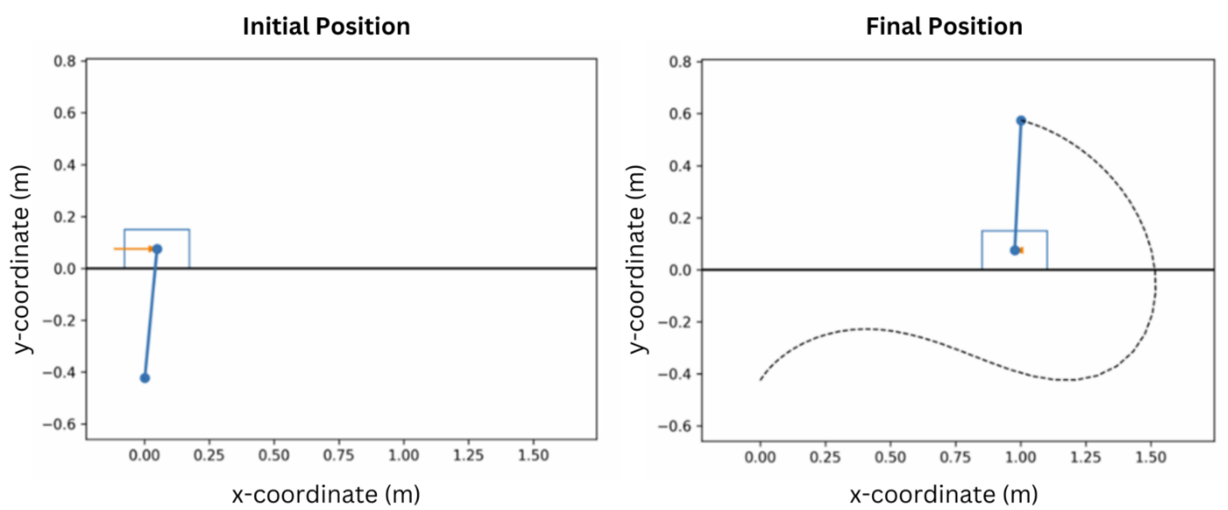


Fig. 1.5. Initial swerve path discovered by GP to launch pendulum to initial position [14]

1.1.4 Foundations of Genetic Programming

GP is a type of evolutionary algorithm that is inspired by the process of natural evolution and genetics in living organisms. The origins of GP can be traced back to the early 1990s, when John Koza formalized the method as an extension of genetic algorithms to evolve actual computer programs, not just fixed-length strings [15]. Since then, GP has been applied across many fields, including symbolic regression, automated design of complex geometries, and even control systems. Just as biological evolution creates adapted life forms over generations (like Darwin's theory of natural selection and survival of the fittest), GP searches for better and better solutions to a given problem by mimicking this process [16]. In GP, a population of possible solutions, usually represented as programs or equations, evolves over time. Each solution is evaluated based on how well it performs a task, like how organisms are judged by their fitness to survive in their environment [17]. The best-performing individuals are selected to "reproduce" and pass on their traits to the next generation.

This reproduction involves several genetic operations. Replication is the simplest, where a solution is directly copied into the next generation without any changes. Crossover is more interesting: it works like genetic recombination in biology, where parts of two parent solutions are combined to form one or more children. This allows useful parts of different solutions to be mixed in hopes of creating an even better one [18]. Mutation introduces random changes into a solution, like random genetic mutations in DNA. These mutations are usually small and help introduce new traits that may be useful later or allow the population to escape from local optima [19]. Crossover and mutation principles are shown in figure 1.6. Over many generations, GP gradually refines its population, keeping the best traits and discarding the less useful ones.

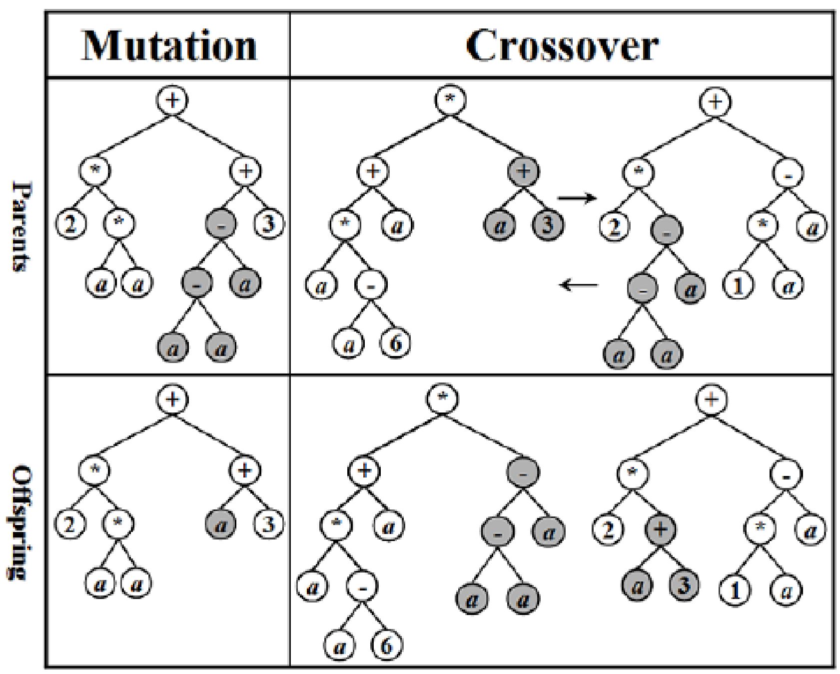


Fig. 1.6. Mutation and crossover in GP [18]

This way of searching for solutions does not require predefined models or structures. It builds them from scratch based on performance. This makes GP a powerful method for finding control laws for complex nonlinear systems, such as those in fluid dynamics, where traditional methods might struggle. A detailed explanation of how these operations is applied in this work to solve this specific research problem is provided in section 2.2.

1.2 Literature Summary

As shown in section 1.1.3, GP demonstrated a substantial advantage in the two study cases of the double inverted pendulum and the cart pendulum systems. Not only did GP achieve a 30% improvement in angle control over the LQR control algorithm for the double pendulum system, but it also demonstrated an impressive initial swerve path for inverting the pendulum to an upright position. Furthermore, GP did so without any explicit system modeling or compromising the system equations by introducing linearization. These two study cases show only a fragment of GP's usability in nonlinear dynamics. Thousands of different nonlinear problems in current existing systems cause either system inefficiencies or inconveniences to users. GP is a steppingstone to resolving such problems.

However, the previous study cases with the inverted pendulums are frankly simple systems, with each comprising two or three degrees of freedom and in principle have less complicated system dynamics than many of the real-world dynamic systems. Despite the improved performance introduced with GP, the previously applied linear control solutions are perfectly capable. In addition, the simplicity of the nonlinear pendulum equations allowed GP to test thousands of control laws over the span of 500 generations. Unless a supercomputer is utilized, this will be difficult to achieve with GP. Therefore, it is also important to find methods to boost GP's performance when running for just a few generations on more complex fluid systems. This research aims to tackle that issue by applying GP to more complex systems where linear control methods do not perform as well.

1.3 Research Goals

This project aims to leverage the strengths of GP to address the challenges posed by drag force acting on nonlinear fluid systems. The fluidic pinball serves as a representative testbed for exploring these challenges. The visual representation of the problem is displayed in figure 1.7. Figure 1.7 shows the ideal steady-state velocity field of the fluidic pinball system as well as the real, chaotic vorticity of the system.

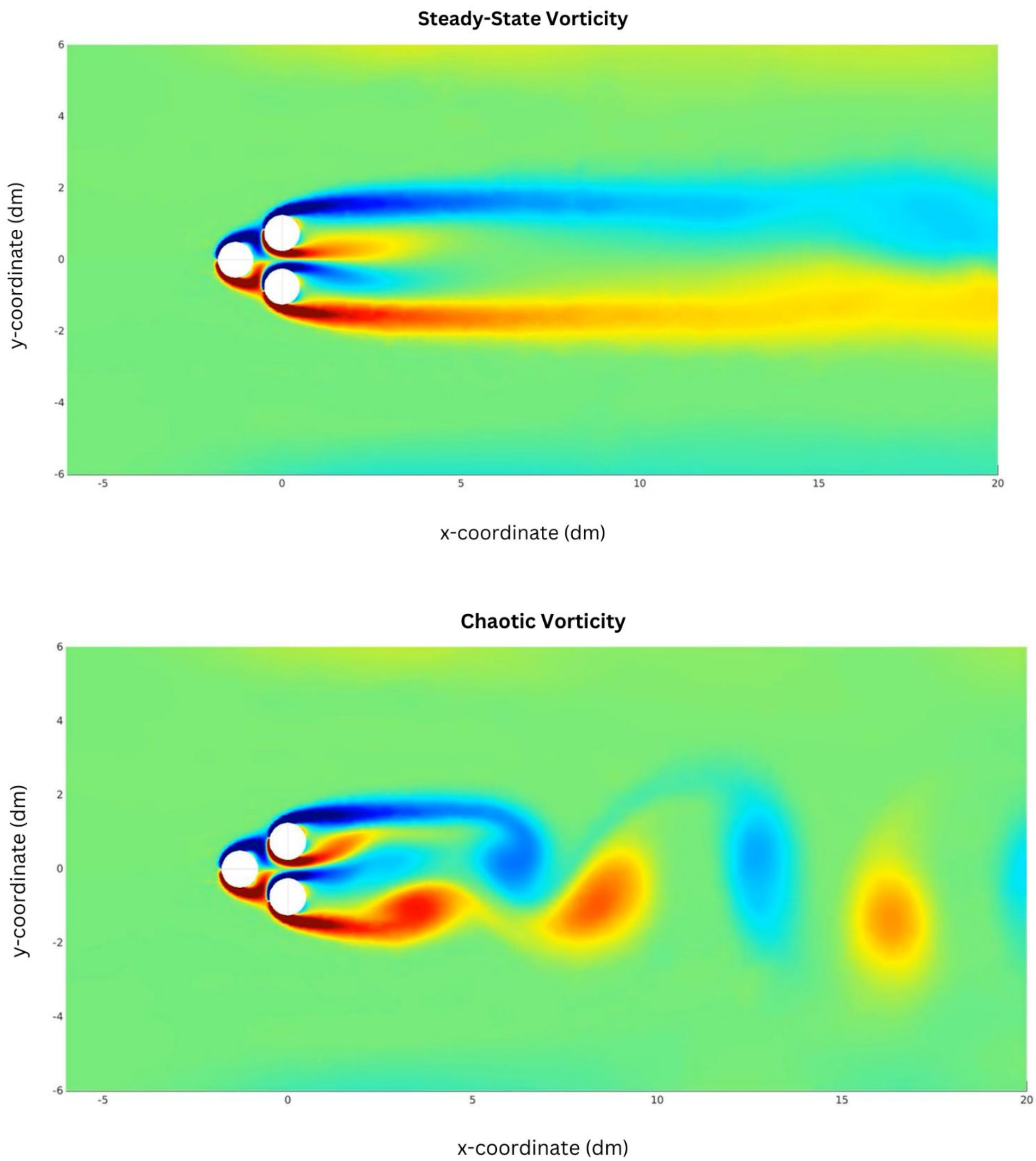


Fig. 1.7. Steady-state vs. chaotic vorticity of the fluidic pinball system

The reduction in drag force and reduction of energy lost is the result of closely matching the real actuated velocity field to the steady velocity field, which is directly reflected in the vorticity diagram. As observed in the chaotic vorticity diagram, high pressure forms upstream due to flow stagnation, and low pressure develops downstream due to flow separation and vortex shedding. This pressure imbalance creates a net force opposing the flow direction, contributing to an increased drag coefficient and drag force. The research must minimize the drag coefficient and drag force with the following objectives:

1. Develop control laws using GP to stabilize the chaotic and nonlinear dynamics of the fluidic pinball to reduce the drag force acting on the system.
2. Demonstrate the advantages of GP in eliminating the need for explicit system modeling and addressing the complexities of real-world fluid systems.

By achieving these goals, the project seeks to advance the field of nonlinear control, specifically, contributing to drag reduction and flow stability.

2. METHODS

This section of the paper thoroughly discusses the tools used and how the research is conducted. It is divided into three subsections to highlight discuss the GP toolkit used for this research, the properties of GP and its method of operation, and later introduces how GP is used on the fluidic pinball.

2.1 OpenMLC-Matlab-2 Toolbox

The OpenMLC-Matlab-2 toolbox plays a central role in this project. It enables the application of GP to evolve control strategies directly from experimental or simulation data. Unlike traditional machine learning methods, which often require extensive datasets and computational resources, OpenMLC-Matlab-2 operates efficiently by leveraging evolutionary optimization principles.

Duriez et al. [20] demonstrated the toolbox's efficacy in optimizing control strategies for a turbulent wake behind a cylinder. Additionally, the toolbox has been applied to economic modeling and weather forecasting, further validating its versatility in addressing nonlinear problems across domains [21-23].

2.2 GP-Based Machine Learning of the Automatic Control Problem

The main architecture adapted when using MLC in dynamical systems is shown in figure 2.1. This model represents the MLC framework using GP [24,25]. It consists of three interconnected components:

1. **Dynamics**: represents the system or process to be controlled. It takes an input $\boldsymbol{w}$, which includes disturbances or reference signals, and outputs the system's state or behavior $\boldsymbol{a}$.
2. **Controller**: This block generates control signals $\boldsymbol{b}$ based on the system's state measure by sensor vector $\boldsymbol{s}$. It uses offline learning through GP to evolve and optimize control law design. GP iteratively improves controller performance by selecting and breeding the most effective individuals.
3. **Cost Function**: This evaluates the system's performance $\boldsymbol{J}$ based on the output $\boldsymbol{z}$ and potentially other criteria. The feedback from the cost function guides the offline learning process to evolve control laws that minimize or optimize the performance index.

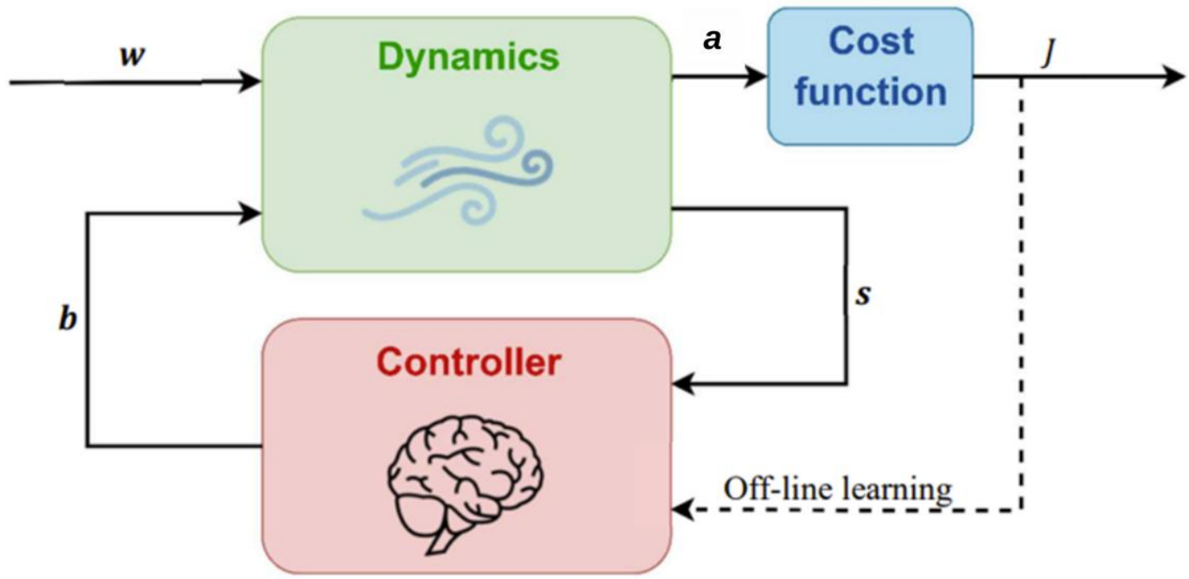


Fig. 2.1. Architecture of MLC in dynamical systems

The adapted GP process consists of several key stages: Exploration, Evaluation, Selection, Genetic Modification, and Convergence. This algorithm is shown in figure 2.2.

Algorithm 1: Genetic Programming Machine Learning Control: GP-MLC

```
Result: K*(t, s), the best individual
Exploration Stage
 | Randomly generate initial population with N_i individuals;
end
Control Law Evolution Cycle
 | while N_gc < N_gp do
 |  | Evaluation Stage
 |  |  | Evaluate current generation N_gc individuals on the dynamical system;
 |  | end
 |  | Selection Stage
 |  |  | Reorganize individuals according to their fitness determined by cost
 |  |  |   function J;
 |  | end
 |  | Genetic Modification Stage
 |  |  | Genetic operations like crossover and mutation are applied to the most-fit
 |  |  |   individuals;
 |  |  | Newly generated offspring is added to the pool, forming the next
 |  |  |   generation N_gc + 1;
 |  | end
 | end
end
Convergence Stage
 | The final individual K*(t, s) is chosen based on its superior performance across
 |   all metrics;
end
```

Fig. 2.2. GP Algorithm

2.2.1 Control Law Framework

The primary goal of this study is to develop an optimal control law for the fluidic pinball system. The control law should be efficient, easy to interpret, and capable of achieving the desired system behavior with minimal complexity. To achieve this, the system dynamics are described in state-space form, which provides a mathematical framework for controlling the system.

As represented by the function diagram in figure 2.1, the system is defined by two essential equations:

State Equation

$$\dot{\boldsymbol{a}} = \boldsymbol{F}(\boldsymbol{a}, \boldsymbol{b}) \tag{2.1}$$

where:

- $\boldsymbol{a} \in \boldsymbol{R}^{\boldsymbol{n}}$ represents the state vector, which describes the current condition of the system.
- $\dot{\boldsymbol{a}}$ is the time derivative of the state vector, indicating how the system evolves over time.

- $\boldsymbol{b} \in \boldsymbol{R^n}$ is the actuation vector, which represents the control inputs applied to the system.
- $\boldsymbol{F}$ is a nonlinear function that defines the system's dynamics, capturing how the state and control inputs influence the system's behavior.

Measurement Equation

$$\boldsymbol{s} = \boldsymbol{H}(\boldsymbol{a}) \tag{2.2}$$

where:

- $\boldsymbol{s} \in \boldsymbol{R^n}$ is the sensor vector, which provides measurements of the system's state.
- $\boldsymbol{H}$ is the measurement function, which maps the state vector $\boldsymbol{a}$ to the sensor outputs $\boldsymbol{s}$.

Control Law

As for the control law, the actuation signal $\boldsymbol{b}$ is determined by a nonlinear control law $\boldsymbol{K}$, which depends on both time $\boldsymbol{t}$ and the sensor measurements $\boldsymbol{s}$. This relationship is expressed as:

$$\boldsymbol{b} = \boldsymbol{K}(\boldsymbol{t}, \boldsymbol{s}) \tag{2.3}$$

The control law $\boldsymbol{K}$ is designed to generate the appropriate actuation signals based on the current state of the system, as measured by the sensors. By allowing explicit time dependence in the control law, the framework can accommodate control strategies that combine sensor feedback with periodic or harmonic forcing, which can be particularly useful for systems with oscillatory or chaotic behavior. This is especially the case in the fluidic pinball system characteristics [26].

Optimization Objective

The goal is to find the optimal control law $\boldsymbol{K}^*(\boldsymbol{t}, \boldsymbol{s})$ that minimizes a predefined cost function $\boldsymbol{J}$. The cost function quantifies the performance of the control law, typically measuring how well the system achieves its desired behavior. The optimization problem can be formally written as [27]:

$$\boldsymbol{K}^*(\boldsymbol{t}, \boldsymbol{s}) = \underset{\boldsymbol{K}(\boldsymbol{t},\boldsymbol{s}) \in \boldsymbol{\mathcal{K}}}{\boldsymbol{argmin}} \, \boldsymbol{J} \, [\boldsymbol{K}(\boldsymbol{t}, \boldsymbol{s})] \tag{2.4}$$

where:

- $\boldsymbol{K}$ represents the space of all possible control laws.
- $\boldsymbol{s}$ is the space of sensor signals.

The optimal control law $\boldsymbol{K}^*(\boldsymbol{t}, \boldsymbol{s})$ is the one that minimizes the cost function $\boldsymbol{J}$ while satisfying the system dynamics described by $\boldsymbol{F}$ and the initial condition $\boldsymbol{a_0}$.

2.2.2 Exploration Stage

The Exploration Stage serves as the starting point of the GP process and aims to ensure diversity in the population of control functions. In this phase, an initial generation of control functions, denoted as $\boldsymbol{N_i}$, is randomly generated. This randomness is key to exploring a wide variety of potential solutions in the search space. By starting with a broad array of candidates, the algorithm avoids premature convergence to suboptimal solutions and enables a more comprehensive search for control laws [28]. The randomness inherent in this stage lays the foundation for future refinement, as it gives the algorithm a diverse set of starting points from which to evolve the control law.

2.2.3 Evaluation Stage

Once the initial population is generated, the Evaluation Stage comes into play. During this stage, each individual in the current generation, referred to as $\boldsymbol{N_{gc}}$, is evaluated for its performance when applied to the dynamical system. The performance of these candidates is quantified using a cost function, $\boldsymbol{J}$, which measures how well each control law manages the system's dynamics [29]. The cost function reflects the difference between the desired and actual system behavior, penalizing deviations from optimal performance. This stage is crucial for identifying the most promising candidates and provides the necessary feedback to guide the evolutionary process.

2.2.4 Selection Stage

Following the evaluation of the individuals, the Selection Stage identifies the most-fit individuals based on their performance in the Evaluation Stage. The ranking of individuals is determined by their fitness, with the top performers being selected for further processing. This process mimics natural selection, where the best-adapted individuals are chosen to propagate their traits into the next generation. Selection ensures that only the strongest solutions move forward, narrowing down the pool of candidates and increasing the likelihood of generating a more effective control law over successive generations.

2.2.5 Genetic Modification Stage

In the Genetic Modification Stage, the selected individuals undergo genetic operations to create offspring for the next generation. The key operations in this stage are replication, crossover, and mutation, denoted by $\boldsymbol{P_r}$, $\boldsymbol{P_c}$, and $\boldsymbol{P_m}$, respectively. Replication is perhaps the simplest and includes completely moving a fit control law to the next generation as a replica. Crossover involves combining the traits of two parent solutions to produce offspring with mixed characteristics, promoting diversity and the possibility of discovering novel solutions.

Mutation, on the other hand, introduces small random changes to an individual's control function, helping to maintain diversity in the population and prevent the algorithm from getting stuck in local optima. Together, these operations drive the evolution of the population, generating new control functions that inherit beneficial traits from their predecessors while introducing new variations to explore [30,31].

The evaluation, selection, and genetic modification stages are all part of the control law evolution cycle. The cycle is the iterative process in which the stages of Exploration, Evaluation, Selection, and Genetic Modification are repeated. After each cycle, the population is refined based on performance, progressively converging toward better control solutions. The cycle continues for a predefined number of generations, denoted as $\boldsymbol{N_{gp}}$. This iterative approach allows the algorithm to gradually improve the control law by continuously adapting and evolving the population based on feedback from the dynamical system. The Control Law Evolution Cycle ensures that the search process is dynamic and progressive, homing in on the optimal solution over time.

2.2.6 Convergence Stage

The Convergence stage marks the end of the GP process. At this point, the iterative evolution has reached a stage where further modifications no longer yield significant improvements, or the specified number of generations, $\boldsymbol{N_{gp}}$, has been reached. The final control law, $\boldsymbol{K}^*(\boldsymbol{t},\boldsymbol{s})$, is selected based on its superior performance in the Evaluation Stage. This control law represents the most effective solution discovered through the GP process and can manage the nonlinear dynamics of the system. The Convergence stage signifies the culmination of the search for an optimal control function, with the final control law providing the best balance of performance and robustness.

This entire process, through its cyclic and iterative nature, ensures the development of a control strategy that can effectively navigate the complexities of nonlinear systems.

2.3 The Fluidic Pinball System

The fluidic pinball system is a well-established benchmark problem in fluid dynamics, characterized by its three rotating cylinders arranged in an equilateral triangle [32]. Each cylinder has a diameter $\boldsymbol{D}$, and the side length of the triangle is $\mathbf{1.5}\boldsymbol{D}$. The system is designed to mimic the chaotic and nonlinear behavior of real-world fluid flows, making it an ideal testbed for exploring control strategies in fluid dynamics [33].

2.3.1 System Configuration and Flow Dynamics

The fluid in the system is assumed to be Newtonian, incompressible, and viscous, with a two-dimensional, uniform, and steady incoming flow. The rotation of the cylinders serves as the primary actuation mechanism, allowing for the manipulation of the flow field.

The actuation signal $\boldsymbol{b} \in \boldsymbol{R}^{\boldsymbol{3}}$ represents the angular velocities of the front, bottom, and top cylinders, respectively. Therefore, actuation signal $\boldsymbol{b}$ will always be comprised of $\boldsymbol{b} = [\boldsymbol{b1}, \boldsymbol{b2}, \boldsymbol{b3}]$. A positive angular velocity rotates a cylinder counterclockwise, while a negative value rotates it clockwise. The no-slip condition on the cylinder boundaries ensures that the flow is redirected based on the actuation.

To monitor the system's response, nine velocity probes are positioned downstream of the cylinders. The sensor data is enriched by incorporating time delays for each sensor signal, providing a more comprehensive representation of the flow dynamics. Equation 2.5 represents the sensor time delays: $\boldsymbol{T_0 = 1}$ represents the natural shedding period, and the index $\boldsymbol{i}$ ranges from $\boldsymbol{1\ to\ 9}$. When referencing any of the $\boldsymbol{n_s = 36}$, it is mentioned as $\boldsymbol{s_{i_s}}$. Therefore, i_s ranges from $\boldsymbol{1\ to\ n_s}$:

$$s_i(t) \qquad s_{i+9}\left(t - \frac{T_0}{4}\right) \qquad s_{i+18}\left(t - \frac{T_0}{2}\right) \qquad s_{i+27}\left(t - \frac{3T_0}{4}\right) \tag{2.5}$$

The Reynolds number for the flow is set to,

$$\boldsymbol{Re = \frac{U_\infty}{v} = 100} \tag{2.6}$$

where ν denotes the kinematic viscosity. The full system configuration is shown in figure 2.3. All dimensions in the figure and system are denoted in decimeters.

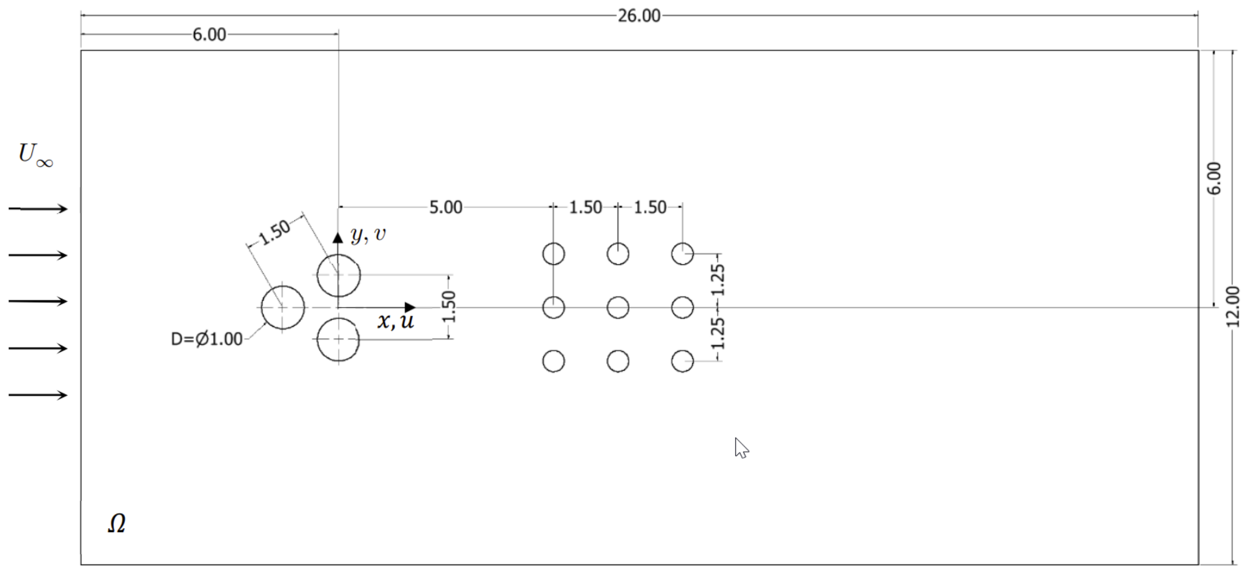


Fig. 2.3. Fluidic pinball computational flow domain Ω with sensor and cylinder configurations

2.3.2 Numerical Simulation Setup

The numerical simulation of the fluidic pinball system is performed using a two-dimensional grid with $\mathbf{4,225}$ triangular elements and $\mathbf{8,633}$ nodes. A time step of $\boldsymbol{dt = 0.1}$ time unit is used, equivalent to $\mathbf{1/10}$ of the natural shedding period. The flow is solved using Direct Numerical Simulation (DNS), and time integration is carried out using the Newton-Raphson method.

The simulation begins with a symmetric steady solution, and the solver is run without actuation for 400 convective time units to allow the transient state to dissipate. Actuation is then applied during the time interval $\boldsymbol{t} \in [\boldsymbol{t_0}, \boldsymbol{t_f}]$, where $\boldsymbol{t_0 = 400s}$ and $\boldsymbol{t_f = 800s}$. The actuation signal is bound to the range $\boldsymbol{b} \in [\mathbf{-5, 5}]$ to simulate the saturation output of real industrial actuators [34].

2.3.3 Control Objectives and Cost Function

The primary control objective is to stabilize the unstable steady wake flow while minimizing actuation energy. This is done by adjusting the rotational speeds of the three cylinders. These cylinder rotations serve as the control input to the dynamic system, also identified as the actuation signal $\boldsymbol{b} \in \boldsymbol{R^3}$. Utilizing the sensor vector $\boldsymbol{s}$ in equation 2.2**,** The actuation signal actively modifies the wake dynamics to reduce flow unsteadiness. Without control, the wake is characterized by periodic vortex shedding, leading to an unstable and highly dynamic flow field. By appropriately tuning the cylinder rotation speeds, we aim to suppress these instabilities and drive the flow toward a steady-state condition. Figure 2.1 presents a functional diagram that visually represents this continuous process.

The drive towards a steady-state solution is guided by the cost function J. The cost function is defined as the sum of two components: $\boldsymbol{J_s}$, which represents the state cost, and $\boldsymbol{J_a}$, which quantifies the actuation power. Specifically:

$$\boldsymbol{J = J_s + \gamma J_a} \tag{2.7}$$

where $\boldsymbol{\gamma = 1}$. The state cost $\boldsymbol{J_s}$ measures the deviation of the actuated velocity field $\boldsymbol{u_b}$ from the steady velocity field $\boldsymbol{u_s}$, using the $\boldsymbol{L_2}$ norm over the computational domain $\boldsymbol{\Omega}$. In simple terms, the control objective is to adjust the cylinder rotations so that the actuated velocity field $\boldsymbol{u_b}$ closely matches the steady velocity field $\boldsymbol{u_s}$, minimizing deviations while keeping actuation energy efficient. A high $\boldsymbol{J_s}$ indicates an unstable flow with significant deviations from the desired steady state. The actuation cost $\boldsymbol{J_a}$ is the time-averaged power required to rotate the cylinders, providing a measure of the energy efficiency of the control strategy.

$$\boldsymbol{J_s = \frac{1}{t_f - t_0}\int_{t_0}^{t_f} \|u_b - u_s\|_{\Omega}^2 dt} \tag{2.8}$$

$$\boldsymbol{J_a = \frac{-1}{t_f - t_0}\int_{t_0}^{t_f} \sum_{i_b=1}^{n_b} \oiint b_{i_b} F_{i_b}^{\theta} ds dt} \tag{2.9}$$

In equations 2.8 and 2.9, The notation $\|*\|_{\boldsymbol{\Omega}}$ represents the $\boldsymbol{L_2}$ norm computed over the flow domain $\boldsymbol{\Omega}$. Additionally, $\boldsymbol{F_{i_b}^{\theta}}$ corresponds to the azimuthal component of the fluid forces acting locally on the cylinder.

2.3.4 GP Configuration and Mathematical Function Library

The control strategy of GP builds a library of candidate functions, which serve as the fundamental building blocks for constructing control laws. These functions include zeroth-order polynomials (constants), first-order polynomials (linear terms), sine functions, and hyperbolic tangent functions applied to each sensor variable. The selection of these functions is deliberate, as they provide a mix of simple and nonlinear transformations that allow the individual to capture both basic and complex relationships between sensor inputs and actuation outputs.

This library forms the basis for generating control functions that determine the actuation strategy. By combining different candidate functions, the control system can represent a wide range of possible control laws, from simple proportional responses to more intricate nonlinear mappings. The diversity of functions ensures that the system has sufficient flexibility to adapt to the fluidic pinball dynamics while maintaining robustness in various flow conditions. The parameters governing the use of these functions are carefully tuned to balance expressiveness and stability, ensuring that the resulting control laws effectively regulate the wake while keeping actuation energy efficient.

The list of candidate functions is portrayed in table 2.1. As for the configuration of GP and the governing parameters for the run. Two different configurations were used, which are displayed in table 2.2. The difference between the two is that the number of generations increases from 8 to 11 generations in the second configuration.

Table 2.1 Suitable GP candidate functions for the fluidic pinball

Candidate Functions	
Polynomial	$*$
Trigonometric	$\boldsymbol{sin}(*)$
	$\boldsymbol{cos}(*)$
Hyperbolic	$\boldsymbol{tanh}(*)$
Rational	$\frac{*}{*}$
Logarithm	$\boldsymbol{log}(*)$
Exponential	$\boldsymbol{e}^{*}$
“*” denotes sensor variables $\boldsymbol{s_1}$ ... $\boldsymbol{s_{36}}$ or polynomial combinations of these sensor variables such as $\boldsymbol{s_2 s_4 s_5}$	

Table 2.2 GP governing parameters for the fluidic pinball

GP Parameter	Configuration Number	Symbol	Value
Number of individuals per generation	1 & 2	$\boldsymbol{N_i}$	50
Total number of generations	1	$\boldsymbol{N_{gp}}$	8
	2		11
Tournament size	1 & 2	$\boldsymbol{N_t}$	7
Probability of replication	1 & 2	$\boldsymbol{P_r}$	0.1
Probability of mutation	1 & 2	$\boldsymbol{P_m}$	0.45
Probability of crossover	1 & 2	$\boldsymbol{P_c}$	0.45

3. RESULTS

This section of the study includes analyzing the initial results after running the offline learning GP algorithm on the fluidic pinball. For clear reference to individual control laws in the results section of the paper, the following format will be used to describe a GP generated control law individual:

$$\boldsymbol{K}_{i_{run}}^{i_{gen},i_{rnk}} \tag{3.1}$$

where $\boldsymbol{i_{gen}}$ is the generation number, $\boldsymbol{i_{rnk}}$ is the ranking of the individual in the generation (1 corresponds to highest ranking), and $\boldsymbol{i_{run}}$ is the GP run number, as multiple runs will be discussed as more GP runs are performed.

3.1 Analysis of GP Runs

3.1.1 Summary of All Runs

Using the fixed configurations presented in table 2.2, three identical runs were performed where GP ran for a full 8 generation span, analyzing over 400 individuals in each run. After that, the second configuration was used to do a single run with an extended 11 generation span. Table 3.1 shows the results of the best individual after the final generation along with its complexity and cost.

Table 3.1 Best individual from each GP run

Run	Best Individual	Complexity c	Cost J	Configuration
1	$\boldsymbol{K}_1^{8,1} = \left[-\mathbf{0.095}\,\boldsymbol{sin}\left(\frac{-\mathbf{3.464}}{\boldsymbol{tanh}(\boldsymbol{tanh}\,(\boldsymbol{sin}(\boldsymbol{s}_8) + \mathbf{3.299}))}\right),\quad \boldsymbol{sin}(\boldsymbol{s}_{17} - \mathbf{1.086}),\quad \mathbf{0.4158}\right]^T$	39	0.2537	1
2	$\boldsymbol{K}_2^{8,1} = [-\mathbf{0.1123},\quad -\mathbf{0.1024},\quad -\mathbf{0.0519}^{(s_{26}+s_{10})}]^T$	34	50	
3	$\boldsymbol{K}_3^{8,1} = [\mathbf{0.0460},\quad \boldsymbol{sin}(\boldsymbol{s}_{17} - \mathbf{1.085}),\quad \mathbf{0.4158}]^T$	9	16	
4	$\boldsymbol{K}_4^{11,1} = \left[\mathbf{0.0460},\quad \boldsymbol{sin}\left(\frac{\boldsymbol{s}_{11} - \mathbf{0.4307}}{-\mathbf{10.535}}\right),\quad \frac{\boldsymbol{e}^{\boldsymbol{tanh}(s_{16}-4.349)}}{\mathbf{4.555}}\right]^T$	41	0.2215	2

An immediate observation to be made is that individuals $\boldsymbol{K}_2^{8,1}$ and $\boldsymbol{K}_3^{8,1}$ failed to converge to a desired cost over the eight-generation-span. Due to GP running on a personal home computer, the number of individuals per generation was limited. This constraint affected the ability to achieve a more optimized control law. Perhaps letting these runs continue for several more generations might have resulted in a desired convergence to the cost $\boldsymbol{J}$. However, they should not be considered in the upcoming discussion concerning controller efficiency as they did not provide the desired outcome. Therefore, the upcoming evaluation of control laws in the next section will only consider individuals $\boldsymbol{K_1^{8,1}}$ and $\boldsymbol{K_4^{11,1}}$, which proved a successful convergence. Even though they used different configurations (with the main difference just being the number of generations run), they both resulted in unique, effective control laws. The next subsection of the report focuses on a detailed analysis of GP runs number 1 and 4 and the effectiveness of the control laws on the fluidic pinball system.

3.1.2 Analysis of Run 1

The first GP run evolved control laws based on the training data and candidate functions presented in table 3.1. The top, evolved control law after the eighth generation of run 1 was:

$$\boldsymbol{K_1^{8,1}} = \left[-\mathbf{0.095}\,\boldsymbol{sin}\left(\frac{-\mathbf{3.464}}{\boldsymbol{tanh}\big(\boldsymbol{tanh}\,(\boldsymbol{sin}(\boldsymbol{s_8}) + \mathbf{3.299})\big)}\right), \quad \boldsymbol{sin}(\boldsymbol{s_{17}} - \mathbf{1.086}), \quad \mathbf{0.4158} \right]^{\boldsymbol{T}} \tag{3.2}$$

The first observation to be made about the control law is the diversity in the input actuation signals. The first actuation signal is a very complex one that would almost never be deducted from an approximate mathematical model of the system while the second is simpler and the third was deemed by GP to be most efficient at the demonstrated linear term.

A detailed examination of the evolutionary process leading to the control law presented in Equation 3.2 reveals several noteworthy observations. As shown in Figure 3.1, the progressive convergence demonstrates significant potential in developing an effective control law for the fluidic pinball system.

One of the important discussions to be made is the relationship between cost and complexity of a control law during the control law evolution cycle. An analysis of the complexity graph yields thought-provoking insights. While the cost function $\boldsymbol{J}$ showed significant improvement, with a reduction of approximately half compared to the unforced dynamical system, the complexity $\boldsymbol{c}$ of the top individual increased substantially between generations 5 and 6. Despite this increase, convergence did not stabilize further, suggesting that additional complexity did not necessarily lead to better performance. In the final generations, the control law complexity exhibited only a minor reduction, indicating that the evolved control laws remained structurally intricate even as cost minimization plateaued. This trend highlights the relationship between increasing complexity and diminishing returns in terms of cost reduction.

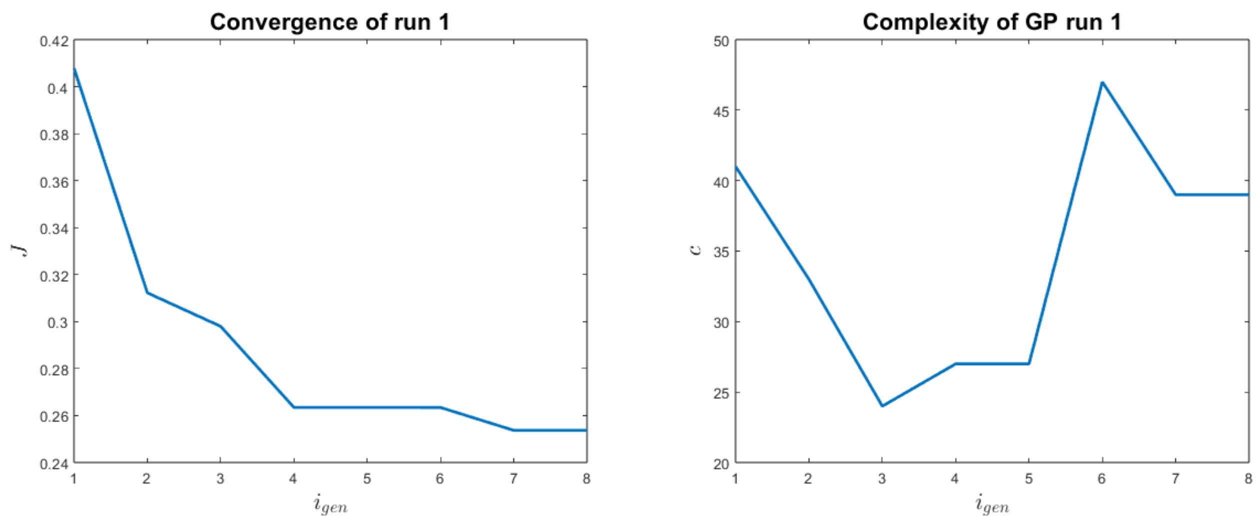


Fig. 3.1. GP run 1 top individual cost and complexity

The rise in control law complexity without useful returns in the convergence of the cost raises an important question: should the complexity of the control law be considered during the selection stage of genetic programming? If an individual exhibits lower complexity while achieving the same cost, it may be advantageous to prioritize it in the tournament ranking. This would not only lead to faster computation by the controller on the real-time system dynamics, but would also make the control law more interpretable for control engineers.

Further observation can be made by comparing the top individuals of generations 7 and 8. The top individual from generation 7 was, if observed carefully, an exact replica of the top individual from generation 8:

$$\boldsymbol{K}_1^{7,1} = \left[-\boldsymbol{0.095 sin} \left(\frac{\boldsymbol{-3.464}}{\boldsymbol{tanh}\big(\boldsymbol{tanh}\,(\boldsymbol{sin}(\boldsymbol{s_8}) + \boldsymbol{3.299})\big)} \right), \quad \boldsymbol{sin}(\boldsymbol{s_{17}} - \boldsymbol{1.086}), \quad \boldsymbol{0.4158} \right]^{\boldsymbol{T}} \tag{3.3}$$

The result makes sense, since replication is one of the GP genetic modification settings we set in the parameters section. However, not only did GP perfectly replicate the top individual in generation 7, but it also failed to produce a more efficient individual in generation 8 through tuning, or further crossover and mutation. This led to the same individual taking the top spot again at the end selection stage of generation 8. Perhaps given more generations to run, the top control law would evolve further, but it raises the concern that GP sometimes gets stuck in a form of premature convergence when converging towards $\boldsymbol{K}^*(\boldsymbol{t}, \boldsymbol{s})$.

3.1.3 Analysis of Run 4

In a major difference to run 1 and configuration 1, GP run 4 used configuration 2 from table 2.2 and ran for a total of 11 generations. This change allows us to observe GP's behavior on a more detailed level as the generations progress and GP is allowed to use more genetic modifications in the genetic modification stage. Unfortunately, the limited computational resources only allowed one run with this configuration, but it achieved a better convergence as compared to run 1, which ran for only 8 generations. The top individual after the 11th generation is shown in equation 3.4:

$$\boldsymbol{K_4^{11,1}} = \left[\mathbf{0.0460}, \quad \boldsymbol{sin}\left(\frac{\boldsymbol{s_{11}} - \mathbf{0.4307}}{\mathbf{-10.535}}\right), \quad \frac{\boldsymbol{e^{tanh(s_{16}-4.349)}}}{\mathbf{4.555}}\right]^{\boldsymbol{T}} \tag{3.4}$$

The first observation to make is the individuality and uniqueness between the top individual in run 4 vs. that of run 1, shown in equation 3.2. While $\boldsymbol{K_1^{8,1}}$ focused on sensor measurement values $\boldsymbol{s_8}$ and $\boldsymbol{s_{17}}$, $\boldsymbol{K_4^{11,1}}$ generated nonlinear functions focused on $\boldsymbol{s_{11}}$ and $\boldsymbol{s_{16}}$. However, considering the number of nonlinear functions used (displayed in the candidate function table 2.1), the sensor vector in equation 2.5, and the possible mathematical computational combinations between them, the possibilities become too complex to make an analysis on such control law parameters. This is precisely the advantage of GP. It is used as effectively as possible to make a detailed analysis of the search space and tries out the different combinations to form a more efficient control law with every evolution step.

Next comes the convergence and complexity of the generated control laws over the course of GP run 4, which are shown in figure 3.2.

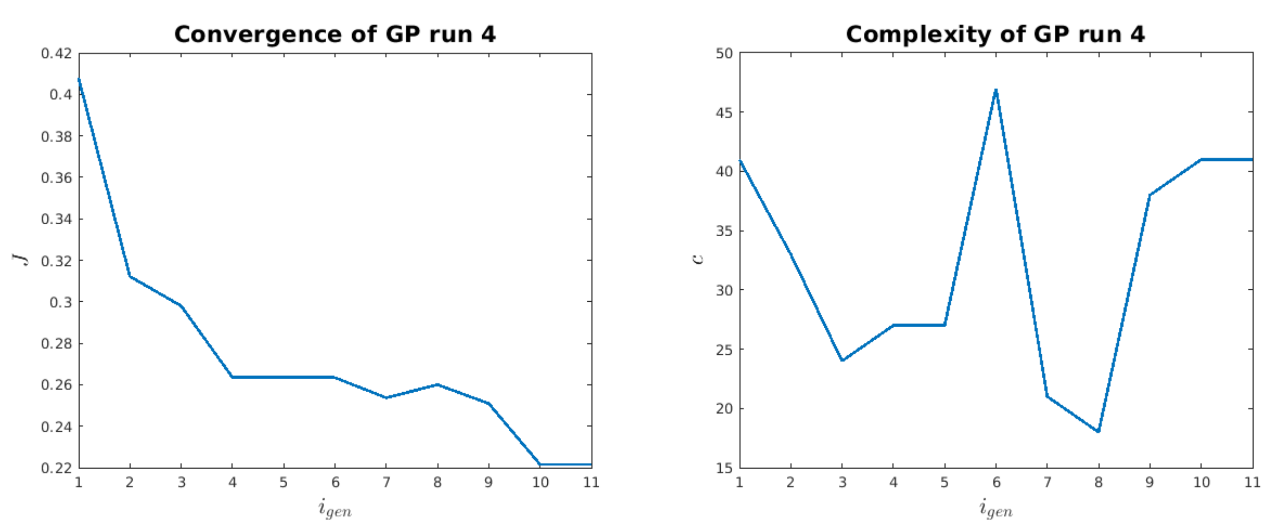


Fig. 3.2. GP run 4 top individual cost and complexity

As shown in figure 3.2, GP showed a somehow steady convergence rate over the 11 generations. However, it exhibited a strange behavior in the convergence between generation 7 and 8. If only by a little bit, the cost made a slight increase between top individual $\boldsymbol{K_4^{7,1}}$ **and** $\boldsymbol{K_4^{8,1}}$. This can only mean that the genetic modification stage of GP failed to replicate the top fit individual, and used crossover and mutation to develop a new individual with lower complexity yet a slightly worsened cost. Considering this research and its purpose, such behavior should be completely avoided and forbidden as the target of the top individual is to reduce the cost to almost 0 as the generations progress. Perhaps as part of the future work, it can be observed whether trading off some of the cost in return for a reduced complexity might result in a more efficient controller for high frequency nonlinear dynamical systems.

3.2 Fitness and Performance Analysis of GP vs. Optimal Linear Control Laws

This subsection of the results focuses on comparing the different properties that make up the fitness of a control law between different control laws generated through different methods.

3.2.1 Cost and Complexity

To put the obtained GP results into perspective and determine whether the control laws introduced performance improvements to the system's response, we must compare them to other control laws specifically designed for the fluidic pinball system with similar input parameters. In their study, Guy and Yiqing et al. demonstrated the stabilization of the fluidic pinball by designing an LQR for the fluidic pinball [35]. After performing fine tuning and comparing its performance to other linear control methods, their final optimal control law was:

$$\boldsymbol{K_{Linear}^{Optimal}} = [\mathbf{1.11207}, \quad \mathbf{-0.20025}, \quad \mathbf{-0.15588}]^{\boldsymbol{T}} \tag{3.5}$$

A fair comparison between the control laws can only be made if they were tested under exactly replicated inputs and conditions. Therefore, the linear control inputs were tested on the identical fluidic pinball system used for the GP runs, and resulted in an expected reduced complexity of $\mathbf{3}$ (due to only linear terms for the control law $\boldsymbol{K}$) and an impressive cost of $\boldsymbol{J} = \mathbf{0.2670}$. A cost $\boldsymbol{J}$ vs. complexity $\boldsymbol{c}$ comparison between $\boldsymbol{K_{Linear}^{Optimal}}$, $\boldsymbol{K_1^{8,1}}$ and $\boldsymbol{K_4^{11,1}}$ is shown in figure 3.3.

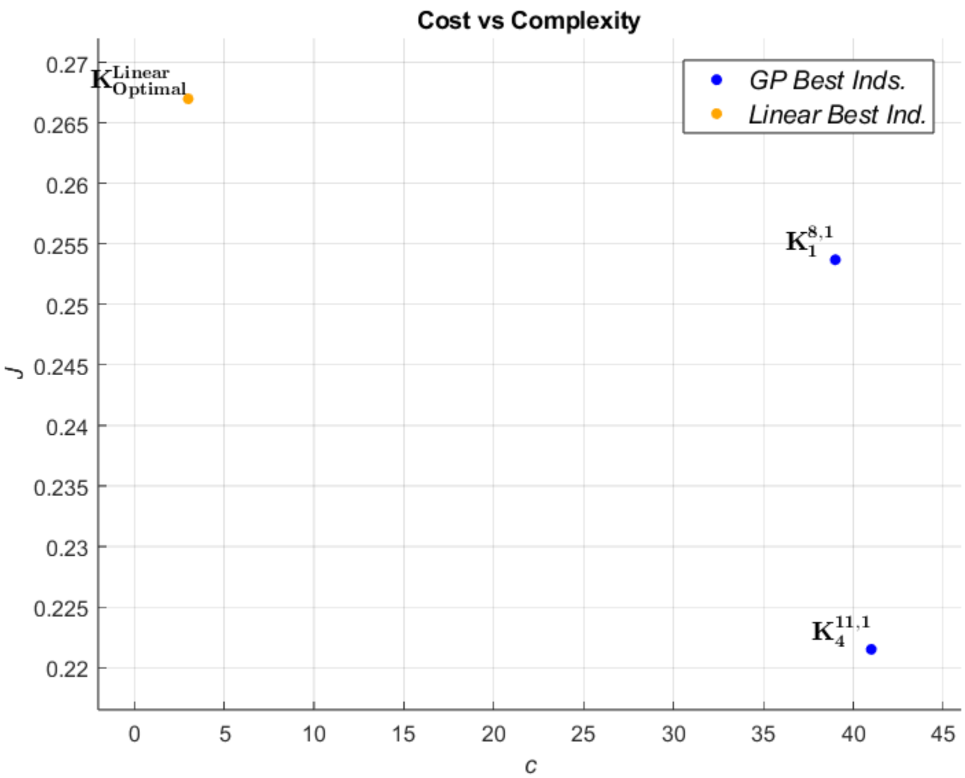


Fig. 3.3. Cost vs. Complexity comparison between best GP and linear control laws

According to figure 3.3, an ideal control law will assume the bottom-left spot in the graph, indicating a minimal cost $\boldsymbol{J}$ and mathematical complexity $\boldsymbol{c}$. As expected from the linear control law, it has the lowest complexity compared to the GP individuals, and as expected from the GP generated control laws, they display better convergence that the optimal linear control law. GP individuals are specifically designed to control the non-simplified, nonlinear model of the system. It also seems like a general rule that the complexity of the top GP individual increases as the generations progress. The more mutation and crossover procedures are performed on top individuals at the end of each generation, the more mathematically complex the nonlinear functions become. However, a successful trade-off is possible. Figure 3.3 shows an impressive convergence in cost over a minor increase in complexity between $\boldsymbol{K}_{\mathbf{1}}^{\mathbf{8,1}}$ and $\boldsymbol{K}_{\mathbf{4}}^{\mathbf{11,1}}$ in only an additional 3 generation span.

3.2.2 System Response

Before analyzing the drag coefficient and the energy consumption difference between the application of the three different control laws, this subsection first examines the flow field response of the system under each control law. The flow response is directly tied to the drag-reduction performance of each control strategy. As discussed in Section 1.3, the closer the actuated velocity field resembles the symmetric steady-state solution shown in Figure 1.5, the lower the resulting drag force on the cylinders. Section 2.3.1 defines the time interval during which the control laws begin to apply actuation to the system, namely $\boldsymbol{t} \in [\boldsymbol{t_0}, \boldsymbol{t_f}]$, where $\boldsymbol{t_0 = 400s}$ and $\boldsymbol{t_f = 800s}$ This delay allows the flow to evolve well into a fully developed unsteady regime, characterized by chaotic vortex shedding, before control is initiated. Starting with the application of the first GP control law $\boldsymbol{K_1^{8,1}}$, the system response is shown in figure 3.4. The figures illustrate the critical period immediately following the onset of actuation, with snapshots taken every 2 seconds.

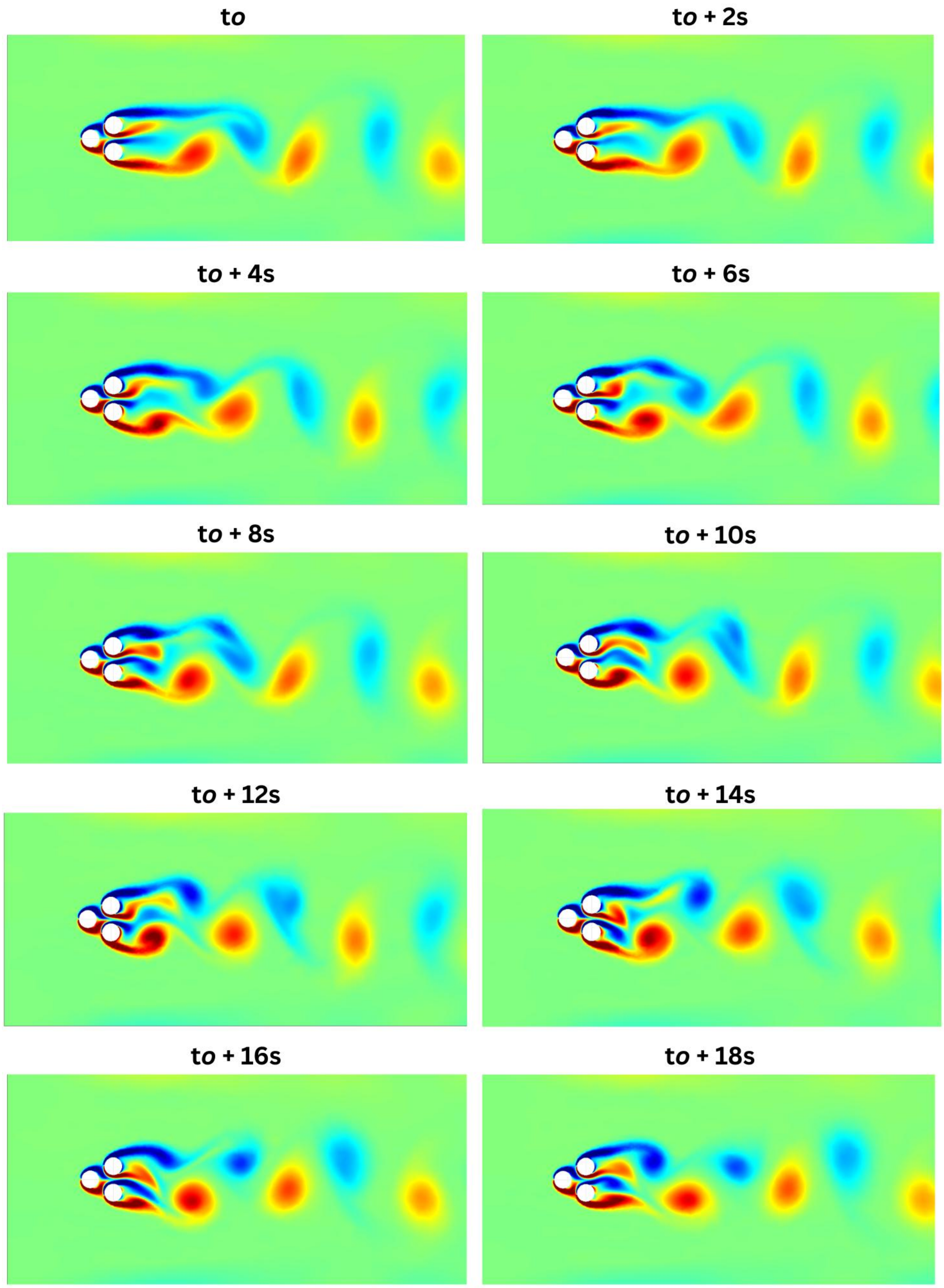


Fig. 3.4. System response under $K_1^{8,1}$

Starting from around time point $\boldsymbol{t_0 + 6s}$ in figure 3.4, it is evident from the flow field that $\boldsymbol{K_1^{8,1}}$ struggles to suppress vortex shedding. Vortical structures continue to be shed throughout the interval up to $\boldsymbol{t_0 + 18s}$. Even though the convergence of this control law reached a minimal cost value of $\boldsymbol{J = 0.2537}$, it seems this convergence was a result of a low actuation cost $\boldsymbol{J_a}$, not a state cost $\boldsymbol{J_s}$, since total cost is a function of both: $\boldsymbol{J = J_s + \gamma J_a}$. This is quite an interesting relationship, as there are limitless possibilities to design a relationship between the two cost values, and perhaps giving the actuation cost $\boldsymbol{J_a}$ a lower weight in the equation than state cost $\boldsymbol{J_s}$ by assigning a lower value than $\boldsymbol{\gamma = 1}$, could shift the optimization toward better flow stabilization at the expense of increased actuation effort. However, this relationship between power consumed by the system vs. power lost from the system will be explained later in the energy analysis section of the results.

Next comes the application of the GP control law $\boldsymbol{K_4^{11,1}}$, for which the system response is shown in figure 3.5.

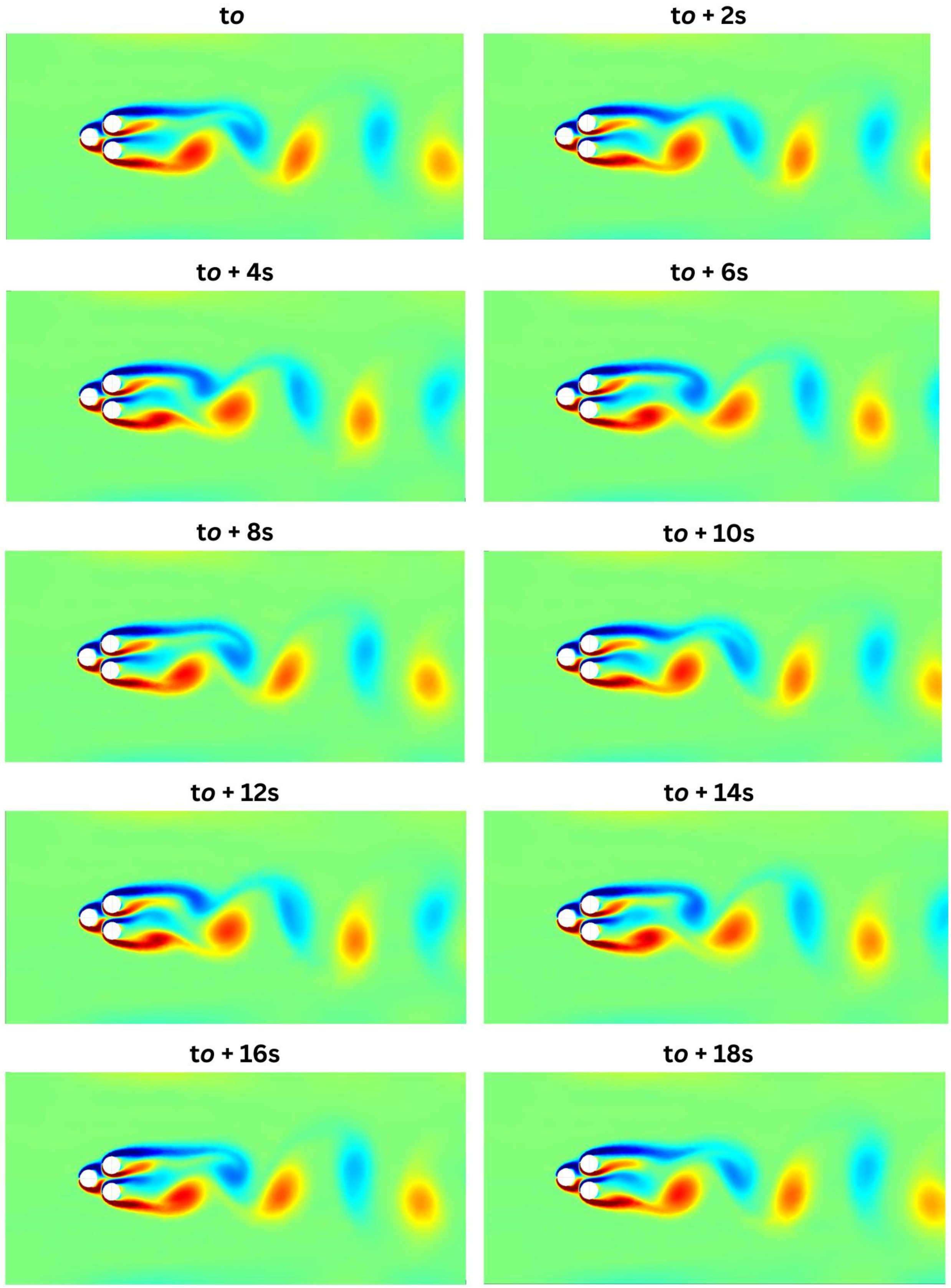


Fig. 3.5. System response under $K_4^{11,1}$

As we immediately notice from an early time-point like $\boldsymbol{t_0 + 4s}$ when comparing figures 3.5 and 3.4, $\boldsymbol{K_4^{11,1}}$ does a much better job at mimicking a steady state velocity field as compared to $\boldsymbol{K_1^{8,1}}$. Firstly, under $\boldsymbol{K_4^{11,1}}$, both the clockwise and the counterclockwise fluid vortices (represented in red and blue, respectively) form further downstream of the cylinders when compared to the system under $\boldsymbol{K_1^{8,1}}$. When vortex shedding occurs closer to the cylinder surfaces, the associated pressure asymmetries contribute to increased drag. Thus, the downstream displacement of vortices indicates improved drag reduction. This improved system response is only natural given that GP run 4 ran for more generations to converge to $\boldsymbol{K_4^{11,1}}$ as compared to run 1. However, it is quite interesting what only a few additional generations can improve in terms of the response of the system.

Finally, figure 3.6 shows the system response under the $\boldsymbol{K_{Linear}^{Optimal}}$ to discuss the differences between GP control laws vs. an optimal linear control law.

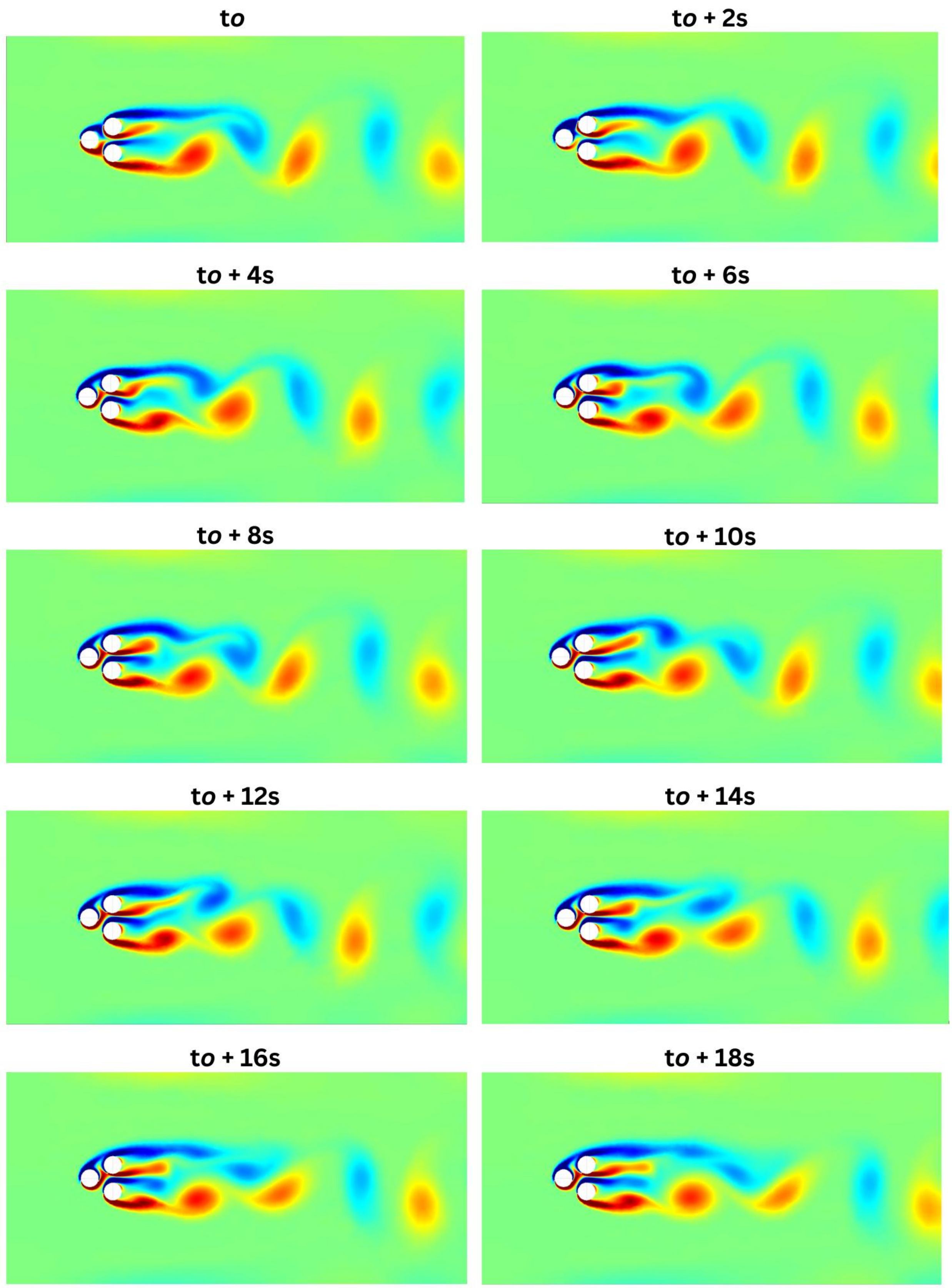


Fig. 3.6. System response under $K_{Linear}^{Optimal}$

As expected from a control law designed by an expert specifically for this fluid dynamic system, $\boldsymbol{K_{Linear}^{Optimal}}$ quickly reaches a great system response as soon as $\boldsymbol{t_0 + 16s}$ and $\boldsymbol{t_0 + 18s}$. By these time-points, the controller effectively suppresses the formation of clockwise vortices in the wake and confines the remaining counterclockwise structures to regions well downstream of the cylinders. This will be further reflected when analyzing the drag coefficient $\boldsymbol{C_d}$ reduction and energy analysis of the system with these different control laws in further subsections of the paper. However, it is immediately clear from the system response in figures 3.4, 3.5, and 3.6 that $\boldsymbol{K_{Linear}^{Optimal}}$ and $\boldsymbol{K_4^{11,1}}$ will outperform $\boldsymbol{K_1^{8,1}}$ when it comes to wake stabilization, regardless of the actuation input each controller required for this system response.

3.2.3 Actuation Power

Recalling equation 2.7, it is important to point out that cost $\boldsymbol{J}$ already hast the component $\boldsymbol{J_a}$ in its computation, entailing that a reduced cost shown in figure 3.3 not only closely matches the actuated velocity field $\boldsymbol{u_b}$ to the steady velocity field $\boldsymbol{u_s}$, but also does so at a reduced actuation power. However, this section will mainly be concerned with power consumption by the system in terms of actuation power $\boldsymbol{P_a}$ and power loss due to drag power $\boldsymbol{P_d}$. By analyzing the energy we introduce into the system in terms of actuation power and the energy lost to drag power, a relationship can be made between the parameters with the introduction of each control law to the system, and the total energy lost can be calculated.

Starting with actuation, the total actuation power $\boldsymbol{P_a}$ in the form of control input vector signal $\boldsymbol{b}$ is:

$$\boldsymbol{P_a = \sum_{i=1}^{3} v_i T_i} \tag{3.6}$$

where:

- $\boldsymbol{i}$ is the cylinder number for a total of 3 inputs for the 3 cylinders.
- $\boldsymbol{v_i}$ is the angular velocity for each cylinder $\boldsymbol{i}$.
- $\boldsymbol{T_i}$ is the torque for each cylinder $\boldsymbol{i}$.

Using equation 3.6 for each time step, the actuation power over time for control laws $\boldsymbol{K_1^{8,1}}$, $\boldsymbol{K_4^{11,1}}$, and $\boldsymbol{K_{Linear}^{Optimal}}$ is displayed in figure 3.7.

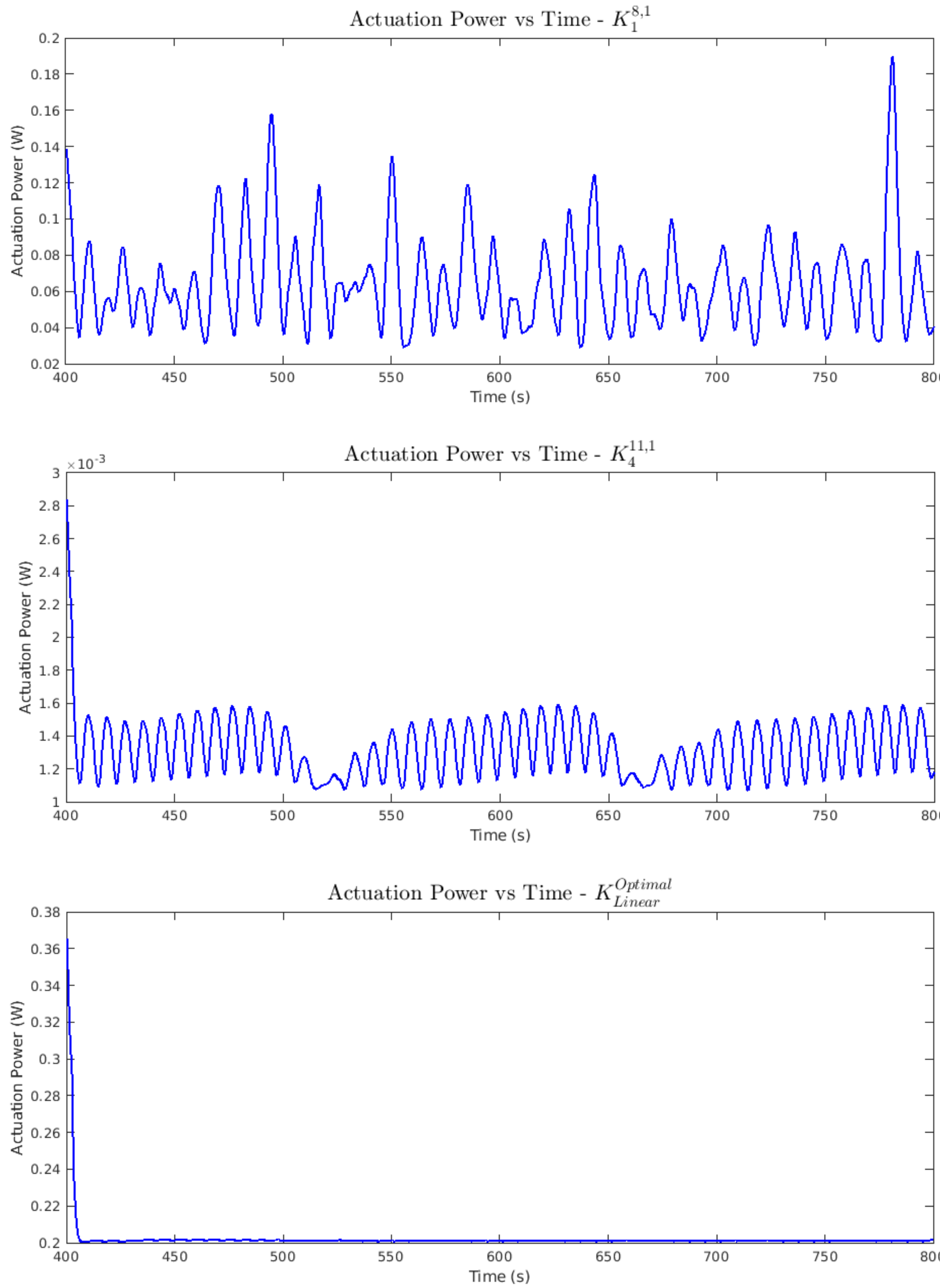


Fig. 3.7. Actuation power of GP and linear control laws

Using equation 3.4 to translate this power into total energy $\boldsymbol{E}$ consumed by the system:

$$E = \int \boldsymbol{P_a(t)dt} \tag{3.7}$$

Table 3.2 summarizes the results of each control law translated from figure 3.7 into mean actuated power and total actuation energy consumed through a summation over time.

Table 3.2 Actuation power and actuation energy consumption

Control Law	Mean Actuation Power (W)	Total Actuation Energy (J)
$\boldsymbol{K_1^{8,1}}$	$\boldsymbol{0.08}$	$\boldsymbol{25.8}$
$\boldsymbol{K_4^{11,1}}$	$\boldsymbol{1.3 * 10^{-3}}$	$\boldsymbol{5.53}$
$\boldsymbol{K_{Linear}^{Optimal}}$	$\boldsymbol{0.21}$	$\boldsymbol{80.9}$

Comparing the input power introduced with each control law: $\boldsymbol{K_1^{8,1}}$ and, $\boldsymbol{K_4^{11,1}}$ have a staggering, mean power and total energy consumed as compared to $\boldsymbol{K_{Linear}^{Optimal}}$, which has the highest energy consumption out of all 3 control laws.

In accordance with the optimization of control laws in relation to the actuation input power, GP did a great job at saving energy, but this can only be deemed noteworthy if it is also translated into a reduction of drag power in the next section. However, what is also noteworthy is the vast energy optimization that $\boldsymbol{K_4^{11,1}}$ achieved over $\boldsymbol{K_1^{8,1}}$ by simply running a few extra generations for optimization. Perhaps the learning curve of GP is not a linear one, and more astounding results are achieved through more generations evaluated than the number of individuals per generation. This gives GP the ability to run more genetic modification procedures such as mutation and crossover, and results in even more robust control laws.

3.2.4 Drag Power

Next comes the consideration of drag power and the energy lost to drag, which is the focus and main problem introduced in the introduction of this research. Consuming lower actuation energy means nothing if not translated into a reduction in energy lost to drag as well.

Specifically for the fluidic pinball system, drag calculation can be done in a simpler manner than the computation done in equation I.1. For the 2-dimensional fluidic pinball system, the drag force $\boldsymbol{F_d}$ can first be calculated through:

$$F_d = \sum_{i=1}^{3} (F_{p,i} + F_{v,i}) \tag{3.8}$$

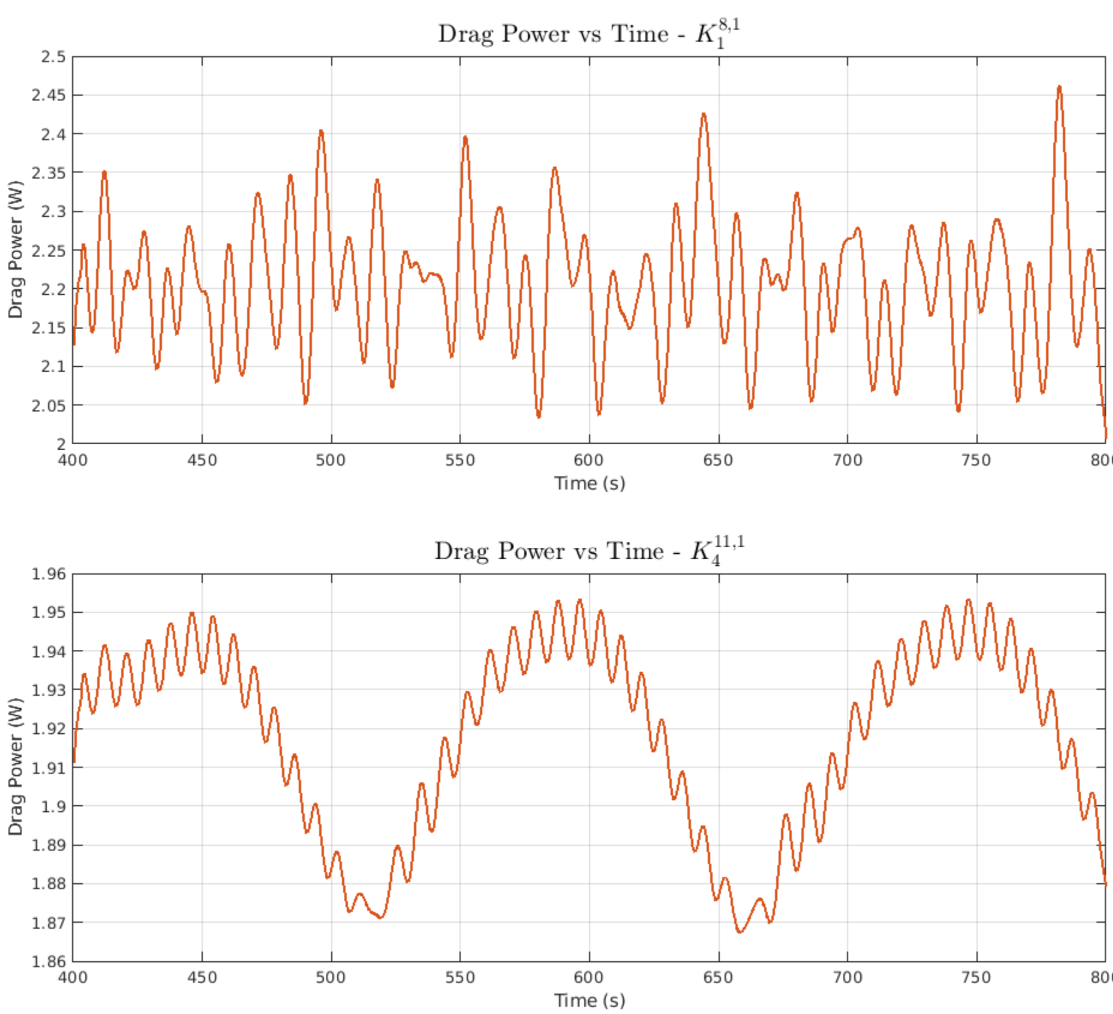
Where:

- $F_{p,i}$ is the pressure force contribution on the surface of cylinder i.
- $F_{v,i}$ is the viscous force contribution on the surface of cylinder i.

Therefore, the drag power P_d can be calculated for every time instance as the product of drag force F_d and the flow velocity component U_x in the drag force direction:

$$P_d = U_x * F_d \tag{3.9}$$

Using this equation, the drag power acting on the fluidic pinball system can be calculated over time, and is displayed in figure 3.8.



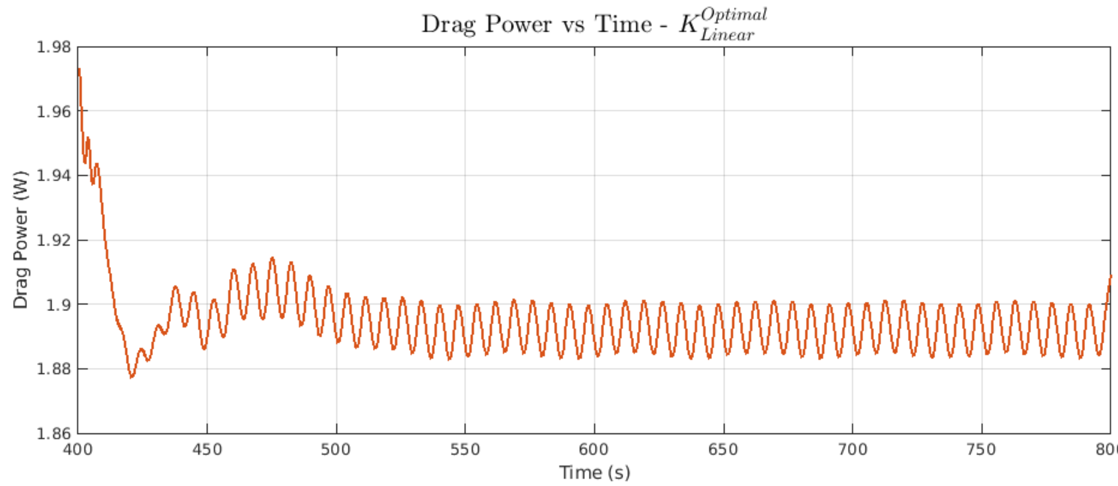


Fig. 3.8. Drag power acting on the system with GP and linear control laws

Using equation 3.7 once more, the total energy lost to drag force can be calculated, and the results of figure 3.8 are summarized in table 3.3.

Table 3.3 Drag power and energy lost to drag

Control Law	Mean Drag Power (W)	Total Energy Lost to Drag (J)
$\boldsymbol{K_1^{8,1}}$	$\mathbf{2.21}$	**881.7**
$\boldsymbol{K_4^{11,1}}$	$\mathbf{1.91}$	$\mathbf{767.1}$
$\boldsymbol{K_{Linear}^{Optimal}}$	$\mathbf{1.89}$	$\mathbf{757.8}$

Table 3.3 raises a lot of important points. After $\boldsymbol{K_4^{11,1}}$ proved much better at power saving actuation over $\boldsymbol{K_{Linear}^{Optimal}}$, it was expected that the energy lost to drag for control law $\boldsymbol{K_4^{11,1}}$ would be quite large compared to $\boldsymbol{K_{Linear}^{Optimal}}$. However, not only does $\boldsymbol{K_4^{11,1}}$ use significantly lower actuation power to control the actuated velocity field, but it also even resulted in an almost similar drag reduction of only $\mathbf{767.1}\ \boldsymbol{J}$ of energy lost.

3.2.5 Drag Coefficient

Recalling the target of this research in terms of drag force reduction, the goal is to reduce the drag force acting on the system by reducing the drag coefficient $\boldsymbol{C_d}$ to a minimum over time. Recalling equation 1.1 in the introduction that relates the drag force to the drag coefficient, and equation 3.9 that converts the drag force to power, we can calculate the drag coefficient for the system over time. In addition to cost $\boldsymbol{J_s}$ used in equation 2.8, the proof of reduction in the drag coefficient will prove the closer matching of the actuated velocity field to the steady state solution, as shown in equation I.2. Integrating equation 3.9 into equation I.1, we can calculate the coefficient as:

$$\boldsymbol{C_d = \frac{2P_d}{\rho A U_x^3}} \tag{3.10}$$

Assuming the fluid used in the simulation is air, we can assume a fluid density $\boldsymbol{\rho = 1.225\ \frac{kg}{m^3}}$. Additionally, looking back at figure 2.3 showing the dimensions and domain of the fluidic pinball system used, the surface Area $\boldsymbol{A}$ can be calculated. In 2D simulations, we usually consider the flow per unit depth for area calculation, instead of the entire wetted area. For the circular cylinders, the projected frontal area is simply the diameter $\boldsymbol{D}$ times a unit depth of 1. With a diameter in decimeters of $\boldsymbol{D = 1\ dm}$ the surface area for all 3 cylinders can be calculated as:

$$\boldsymbol{A = 3 * (D * 1) = 0.3\ m^2} \tag{3.11}$$

Using the calculated drag power $\boldsymbol{P_d}$ and velocity component $\boldsymbol{U_x}$, the mean drag coefficient from the application of the 3 different control laws of $\boldsymbol{K_1^{8,1}}$, $\boldsymbol{K_4^{11,1}}$, and $\boldsymbol{K_{Linear}^{Optimal}}$. The calculated coefficient is shown is shown in table 3.4 and is graphed against the actuation power induced by each control law in figure 3.9.

Table 3.4 Mean drag coefficient with different control laws

Control Law	Mean Drag Coefficient $\boldsymbol{C_d}$
$\boldsymbol{K_1^{8,1}}$	$\mathbf{0.188}$
$\boldsymbol{K_4^{11,1}}$	$\mathbf{0.163}$
$\boldsymbol{K_{Linear}^{Optimal}}$	$\mathbf{0.161}$

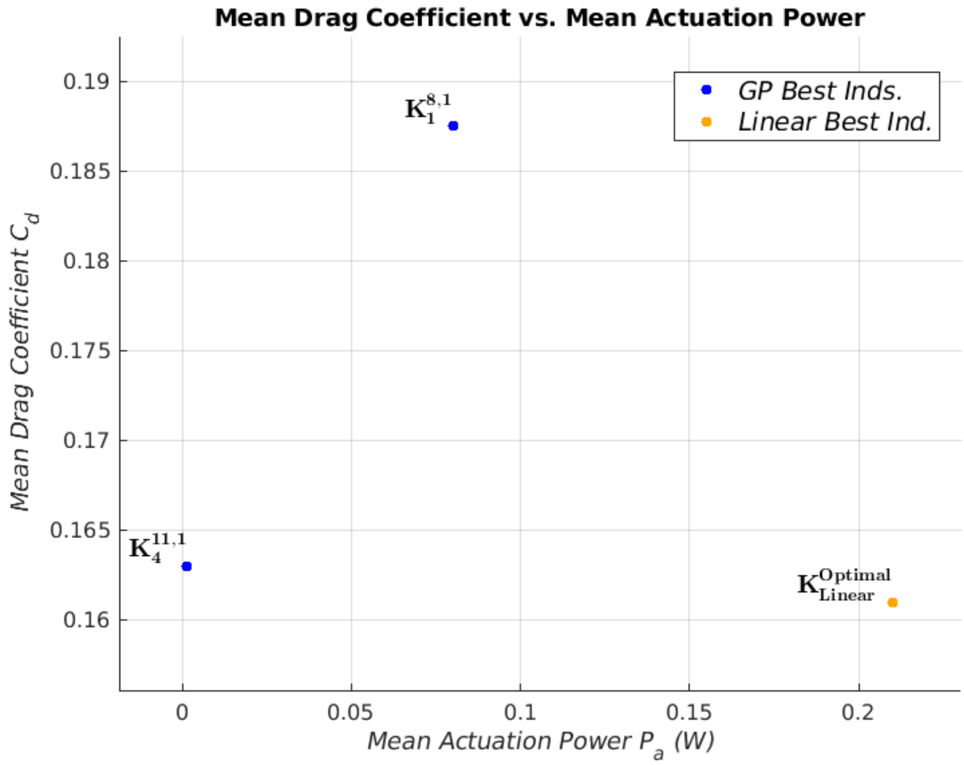


Fig. 3.9. Mean drag coefficient vs. mean actuation power comparison between best GP and linear control laws

Table 3.4 clearly shows the linear control law $\boldsymbol{K_{Linear}^{Optimal}}$ achieved the best/minimum drag coefficient. However, figure 3.6 provides us with more insight and shows this was achieved at a much higher control power. The expended actuation energy is a critical part of this research. Since the aim is to reduce the overall energy wasted due to drag, requiring much energy to do so will only lead to the same wasted resources. On the other hand, $\boldsymbol{K_4^{11,1}}$, achieved an almost similar drag coefficient, but did so at a much lower power. This will be further reflected and explained in the total energy analysis in the next section of the paper.

3.2.6 Energy Analysis

Finally, the total energy consumption calculations are compared to the unforced system (with no control) to show the energy saved by the introduction of each control law. Table 3.5 summarizes the results from this research through the use of GP and compares these results with the unforced system as well as the linear control law $\boldsymbol{K_{Linear}^{Optimal}}$. In addition, the total power consumed can be expressed as the sum of the actuation power $\boldsymbol{P_a}$ drag power $\boldsymbol{P_d}$, and can be integrated over time to find the total energy lost in the system for each control law.

$$P_{total} = P_a * P_d \tag{3.12}$$

Table 3.5 Summary of energy consumption and energy loss with different control laws

Control Law	Actuation Energy (J)	Energy Lost to Drag (J)	Total Energy Lost (J)
Unforced System (No Control)	$\mathbf{0}$	$\mathbf{1109.7}$	$\mathbf{1109.7}$
$\boldsymbol{K_1^{8,1}}$	$\mathbf{25.8}$	**881.7**	$\mathbf{907.5}$
$\boldsymbol{K_4^{11,1}}$	$\mathbf{5.53}$	$\mathbf{767.1}$	$\mathbf{772.63}$
$\boldsymbol{K_{Linear}^{Optimal}}$	$\mathbf{80.9}$	$\mathbf{757.8}$	$\mathbf{837.9}$

To summarize the results in table 3.5, control laws $\boldsymbol{K_1^{8,1}}$ and $\boldsymbol{K_4^{11,1}}$ followed GP's idea of an ideal control law, which we expressed to GP in our cost function $\boldsymbol{J}$ in equation 2.7 as the balance between the steady actuated velocity field and the minimization of actuation power: $\boldsymbol{J = J_s + \gamma J_a}$. GP control laws showed great promise in reducing the total actuation energy as compared to the optimal linear control law $\boldsymbol{K_{Linear}^{Optimal}}$. However, $\boldsymbol{K_1^{8,1}}$ did not properly minimize the energy lost to drag as compared to the energy lost to the unforced system and how it was minimized by $\boldsymbol{K_{Linear}^{Optimal}}$. The original goal of the research was to prove GP's superior performance to linear control methods in minimizing drag, which $\boldsymbol{K_1^{8,1}}$ failed to do. On the other hand, $\boldsymbol{K_4^{11,1}}$ displayed this superiority splendidly. It minimized the actuation energy by almost $\mathbf{93}\%$ as compared to the linear law $\boldsymbol{K_{Linear}^{Optimal}}$, while maintaining almost the same level of energy lost. This difference is amplified even more when looking at the total energy lost in each system, with $\boldsymbol{K_4^{11,1}}$ taking the top spot with minimal energy consumed over the minimized 400 second experiment window. Such is the advantage of implementing GP in such nonlinear dynamical systems, and more improvements can be made and are discussed in the next section of the paper.

3.3 Future Work

After finalizing the obtained results, this subsection of the paper focuses on the elements of the research that should be considered for future analysis as part of improving GP for the control of nonlinear dynamical systems. The following subtopics have proven their relevance to obtaining better results as part of future studies.

- **Number of Individuals vs Number of Generations:** As observed between the performance of $\boldsymbol{K_1^{8,1}}$ and $\boldsymbol{K_4^{11,1}}$, a slightly increased number of additional generations can result in a drastic difference in energy saved. The convergence of GP control laws is not defined by a fixed method. It is very difficult to determine how many generations must pass before a successful convergence is reached. However, there is also the number of individuals per generation, and having more individuals in a single generation can widen the search space before GP starts with the genetic modification stage. There must exist a middle ground between balancing between these two parameters.
- **Cost vs. Complexity:** the relationship between cost and complexity was discussed in detail as part of the results analysing the generated control laws. Complexity is an issue when using GP because individuals become overly complex as generations progress. Finding a method to balance between these two important parameters can help GP obtain even more impressive results.
- **Candidate Function Selection:** The candidate functions shown in table 2.1 are an important aspect for each GP run but were not given much attention or modified throughout this research. In fact, GP runs 2 and 3 mainly failed to converge due to the unsuitable candidate functions that were used to construct control laws for the first generation. Perhaps more detailed research performed regarding candidate function selection can boost the energy gap even further.
- **Comparison with Other Nonlinear Control Methods:** In this research, the generated GP control laws were compared to an optimal linear control law that was developed specifically for this system. Nonetheless, it would be interesting to compare these results with other nonlinear control methods such as neural networks and find the advantages of each approach.
- **Optimization of Different System Output:** As previously mentioned the cost function used by GP can be manipulated to converge to whichever system output parameter desired. This research focused on the actuated velocity field as well as actuation power. However, it would be interesting to attempt to tweak the cost function in favour of one of the two outputs or even converge to other outputs such as the fluctuation energy or lift fluctuation reduction.
- **Different Dynamical Systems:** Finally, perhaps the final step of improving GP is to expand its usefulness to beyond the fluidic pinball and apply it to even more complex dynamical systems. The aerospace industry seems like a perfect candidate with overly

complex models that are difficult to linearize and model. However, this step can only be made once we have a better understanding of all the previously mentioned points and relationships.

CONCLUSION

Reiterating the project tasks discussed in the introduction:

1. **Drag Reduction:** The primary task and objective of reducing drag force through the minimization of drag coefficient $\boldsymbol{C_d}$ was successfully achieved using GP. The evolved controllers demonstrated comparable or improved performance in reducing $\boldsymbol{C_d}$ relative to traditional linear control methods. This was achieved with lower actuation power, highlighting GP's efficiency in handling nonlinear systems like the fluidic pinball. These findings are supported by the data in Figure 3.9 and Table 3.5.
2. **Model-Free Control Development:** GP eliminated the need for explicit system modeling by directly using feedback from the real-time dynamics of the fluidic pinball system. Controllers were evolved entirely through the GP evaluation cycle without any prior knowledge of the system's governing equations, proving the feasibility and efficiency of this model-free approach.
3. **Nonlinear Feedback Adaptation:** The GP-based controllers were developed without relying on linearized approximations, allowing them to adapt directly to the nonlinear wake behavior of the system. This enabled more accurate control of vortex shedding and contributed to drag reduction. The method's success suggests and iterates the high potential for scaling GP-based control to industrial applications.

LIST OF LITERATURE

[1] L. C. Lee, "Nonlinear Control Systems and Applications," *IEEE Control Systems Magazine*, vol. 12, no. 4, pp. 22-32, 1995.

[2] J. Smith, A. Lee, and M. Johnson, "Vortex Shedding and Its Impact on Flow Control," *Journal of Fluid Mechanics*, vol. 24, pp. 145-160, 2001.

[3] R. Brown and E. Wang, "The Role of Vortex Shedding in Flow Instabilities," *AIAA Journal of Fluid Dynamics*, vol. 19, pp. 78-85, 1999.

[4] J. H. Bebernes, "Microfluidic Systems: Modeling and Simulation of Nonlinear Dynamics," *Journal of Microfluidics and Nanofluidics*, vol. 12, no. 2, pp. 233-249, 2015.

[5] Lawrence Livermore National Laboratory, "Study on fuel consumption reduction in trucks due to improved aerodynamics," *Alternative Fuels Data Center*, U.S. Department of Energy, 2021.

[6] A. R. Ghasemi, M. M. Alavi, and M. R. Yousefi, "Study and simulation of the fuel consumption of a vehicle with respect to ambient temperature and weather conditions," *ResearchGate*, 2019.

[7] A. V. Nistor, R. G. Parker, and C. D. Bishop, "Optimization Strategies for Fluidic Systems: Genetic Programming and Other Approaches," *Control Systems and Dynamics*, vol. 45, pp. 99-110, 2019.

[8] A. V. Nistor, R. G. Parker, and C. D. Bishop, "Optimization Strategies for Fluidic Systems: Genetic Programming and Other Approaches," Control Systems and Dynamics, vol. 45, pp. 111-130, 2019.

[9] A. T. Smith, R. G. Curtis, and J. F. Morris, "PID Control for Wake Stabilization in Cylindrical Flow Systems," *IEEE Transactions on Control Systems Technology*, vol. 20, no. 1, pp. 47-58, 2013.

[10] M. Thompson and J. F. Lee, "Challenges in Linearizing Nonlinear Fluid Systems," *Systems and Control Letters*, vol. 41, no. 2, pp. 47-58, 2017.

[11] A. S. Smith and B. L. Jones, "A Comparison of Genetic Algorithms in Optimizing Controllers for Inverted Pendulum," *International Journal of Control Systems*, vol. 33, no. 2, pp. 145-158, 2024.

[12] T.-T.-D. Le, T.-A. Do, Q.-T. Le, Q.-T. Tran, et al., "A Comparison of Genetic Algorithms in Optimizing Controllers for Inverted Pendulum," *Robotica & Management*, vol. 28, no. 2, pp. 21-27, Jan. 2023.

[13] M. W. White, "Stabilization of the Double Pendulum Using Genetic Programming," *Journal of Control Engineering*, vol. 22, pp. 33-45, 2008.

[14] R. Patel and H. K. Ford, "Applying Genetic Programming to the Cart-Pole System," *Journal of Mechanical Engineering*, vol. 17, pp. 150-160, 2009.

[15] J. R. Koza, Genetic Programming: On the Programming of Computers by Means of Natural Selection, MIT Press, 1992.

[16] W. Banzhaf, P. Nordin, R. E. Keller, and F. D. Francone, Genetic Programming: An Introduction: On the Automatic Evolution of Computer Programs and Its Applications, Morgan Kaufmann, 1998.

[17] R. Poli, W. B. Langdon, and N. F. McPhee, A Field Guide to Genetic Programming, Lulu.com, 2008.

[18] S. Luke, Essentials of Metaheuristics, Lulu, 2013. [Online]. Available: https://cs.gmu.edu/~sean/book/metaheuristics/

[19] M. O'Neill and C. Ryan, Grammatical Evolution: Evolutionary Automatic Programming in an Arbitrary Language, Springer, 2003.

[20] D. Duriez, P. L. Johnson, and M. B. A. Kharbat, "Optimization of Fluid Flow Control using OpenMLC-Matlab-2," *Journal of Computational Fluid Dynamics*, vol. 13, pp. 200-220, 2014.

[21] C. Wang, L. Zhao, and Z. B. Hall, "Applications of Genetic Programming in Energy Systems Modeling," *Energy Modeling Journal*, vol. 48, pp. 102-118, 2016.

[22] T. Quade, "Genetic Programming in Turbulent Flow Systems," *Flow Control Techniques Journal*, vol. 34, no. 3, pp. 145-159, 2012.

[23] S. H. Allen and R. B. Moore, "GP in Weather Forecasting: A Comparative Study," *Artificial Intelligence in Engineering*, vol. 40, pp. 22-31, 2015.

[24] A. V. Nistor, "Machine Learning and Genetic Programming in Control," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 235-245, 2017.

[25] B. Y. Liu, "Application of Genetic Programming for Optimization of Control Laws in Dynamic Systems," *Automation and Control Engineering*, vol. 30, pp. 67-81, 2018.

[26] R. S. Kim, "Nonlinear Control of Fluidic Systems using Genetic Programming," *Proceedings of the International Conference on Control*, vol. 11, pp. 134-145, 2019.

[27] D. J. Lewis and S. L. Brown, "Genetic Programming for Control Systems Optimization," *Journal of Computational Control*, vol. 32, pp. 44-56, 2020.

[28] L. O. Yang, "Exploring the Role of Exploration in Genetic Algorithms," *Computational Intelligence*, vol. 18, pp. 129-140, 2018.

[29] J. P. Robson, "Evaluation of Control Strategies for Nonlinear Systems," *Mathematics of Control Systems*, vol. 14, pp. 109-121, 2017.

[30] L. L. Sun and M. J. Reed, "Nonlinear Dynamics and Control of Fluid Systems," *Journal of Fluid Engineering*, vol. 15, pp. 95-105, 2011.

[31] G. F. Mendez and L. P. Hartman, "Fluid Mechanics for Control System Optimization," *Journal of Advanced Fluid Mechanics*, vol. 17, no. 2, pp. 56-68, 2016.

[32] L. R. Pastur, N. Deng, M. Morzyński, and B. R. Noack, "Reduced-order modeling of the fluidic pinball," *arXiv preprint arXiv:2104.05104*, 2021.

[33] R. J. Reiner and D. T. Johnson, “Improving Fluid System Performance Using Advanced Control Algorithms,” *Journal of Control Theory and Applications*, vol. 27, no. 4, pp. 112-124, 2018.
[34] A. V. Nistor, R. G. Parker, and C. D. Bishop, “Optimization Strategies for Fluidic Systems: Genetic Programming and Other Approaches,” *Control Systems and Dynamics*, vol. 45, pp. 99-110, 2019.
[35] G. Y. Cornejo Maceda, Y. Li, F. Lusseyran, M. Morzyński, and B. R. Noack, “Stabilization of the Fluidic Pinball” Fluid Dynamics and Control, vol. 58, 2021.
[36] Movotiv Research Group, “Statistics on Mobility and Human Attention,” Movotiv, 2025.