



Czech Technical University in Prague
Faculty of Electrical Engineering
Department of Control Engineering

Control of Magneto hydrodynamic Flow: Integration of Electronics and Measurement

Bachelor's thesis

Šimon Pecháček

Supervisor
Ing. Jiří Zemánek, Ph.D.

Academic year 2023/2024



BACHELOR'S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Pecháček Šimon** Personal ID number: **508458**
Faculty / Institute: **Faculty of Electrical Engineering**
Department / Institute: **Department of Control Engineering**
Study program: **Cybernetics and Robotics**

II. Bachelor's thesis details

Bachelor's thesis title in English:

Control of Magnetohydrodynamic Flow: Integration of Electronics and Measurement

Bachelor's thesis title in Czech:

Řízení magnetohydrodynamického proudění: integrace elektroniky a měření

Guidelines:

The aim of the project is to complete and document the electronic and software parts of the magnetohydrodynamic (MHD) flow control system. The system consists of the electrode array, the electronics for adjusting the voltage across the electrodes, the velocity field measurement, and the control software. The project will also include experimental work motivated by creating a dataset capturing the dependence of the flow on the MHD force and designing a simple control of the position of an object or multiple objects on the surface.

Tasks:

- To complete and document the electronics for setting the voltage on the electrodes in a magnetohydrodynamic system. Implement the measurement of electric current.
- Create a user interface to control electrode voltages and coil currents manually. Consider incorporating velocity field visualization.
- Finalize and document hardware and software to perform velocity field measurements in the magnetohydrodynamic system using Particle Image Velocimetry (PIV).
- Perform velocity profile measurements under different operating conditions and create a dataset capturing the evolution of velocity profiles over time for different current and voltage settings. Also, consider measuring the effect of the fluid on the magnetic field.
- Develop a simple heuristic strategy to control the position of an object (or multiple objects) placed on the surface of a magnetohydrodynamic system.

Bibliography / sources:

- [1] R. J. Adrian and J. Westerweel. Particle image velocimetry. Cambridge university press, 2011.
[2] S. L. Brunton and J. N. Kutz, Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control, 2nd ed. Cambridge: Cambridge University Press, 2022.

Name and workplace of bachelor's thesis supervisor:

Ing. Jiří Zemánek, Ph.D. Department of Control Engineering FEE

Name and workplace of second bachelor's thesis supervisor or consultant:

Date of bachelor's thesis assignment: **16.02.2024** Deadline for bachelor thesis submission: **24.05.2024**

Assignment valid until: **21.09.2025**

Ing. Jiří Zemánek, Ph.D.
Supervisor's signature

prof. Ing. Michael Šebek, DrSc.
Head of department's signature

prof. Mgr. Petr Páta, Ph.D.
Dean's signature

III. Assignment receipt

The student acknowledges that the bachelor's thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the bachelor's thesis, the author must state the names of consultants and include a list of references.

Date of assignment receipt

Student's signature

Declaration of authorship

I hereby declare that this thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

Prague, May 24, 2024

Šimon Pecháček

Acknowledgments

First, I would like to thank my supervisor, Ing. Jiří Zemánek, PhD., for his expertise and advice. I would also like to thank Adam Uchytíl for his collaboration on this project and for giving me useful tips. Next, I thank Krištof Pučejdl for proofreading my thesis and helping me with mechanical construction. Last but not least, I am grateful to my parents and my girlfriend for supporting me throughout this thesis.

Abstract

This bachelor's thesis aims to develop an experimental platform for the feedback control of magnetohydrodynamic (MHD) fluid flow. It details the design and construction of the hardware, including two control boards. These boards regulate the voltage on electrodes or the current flowing through them. The platform induces the flow of conductive fluids using electric and magnetic fields, the fundamental principle of MHD. A Particle Image Velocimetry (PIV) algorithm is implemented to measure the induced flow. Utilizing this experimental platform and the PIV algorithm, heuristic control algorithms are developed for object manipulation with MHD flow and for controlling steady-state MHD flow and its mean velocity. The thesis also presents the results of preliminary experiments conducted using this platform.

Keywords: Magnetohydrodynamics, MagMan, Hardware development, Electrodes, Particle Image Velocimetry, Heuristic control

Abstrakt

Cílem této bakalářské práce je vyvinout experimentální platformu pro zpětnovazební řízení magnetohydrodynamického (MHD) proudění tekutin. Podrobně popisuje návrh a konstrukci hardwaru, včetně dvou řídicích desek. Tyto desky regulují napětí na elektrodách nebo jimi protékající proud. Platforma vyvolává proudění vodivých tekutin pomocí elektrických a magnetických polí, což jsou základní principy MHD. K měření vyvolaného toku je implementován Particle Image Velocimetry (PIV) algoritmus. S využitím experimentální platformy a PIV algoritmu jsou vyvinuty heuristické řídicí algoritmy pro manipulování s objekty pomocí MHD proudění a pro řízení ustáleného stavu MHD proudění a jeho průměrné rychlosti. Práce také představuje výsledky předběžných experimentů s touto platformou.

Klíčová slova: Magnetohydrodynamika, MagMan, Vývoj Hardware, Elektrody, Particle Image Velocimetry, Heuristické řízení

Contents

Declaration of authorship	v
Acknowledgments	vii
Abstract	ix
Abstrakt	xi
Contents	xiii
Nomenclature	xv
1 Introduction	1
1.1 Motivation & Goals	1
1.2 My Contribution	1
1.3 Structure	2
1.4 Magnetohydrodynamics	2
2 Platform	5
2.1 Camera	6
2.2 The MagMan Platform	6
2.3 Voltage Control Electrode Board	7
2.4 Current Control Electrode Board	13
2.5 Electrodes & Electrolytes	20
2.6 Summary	22
3 Flow Measurement	23
3.1 Principle	23
3.2 Evaluation	25
3.3 Histogram equalization	27
3.4 Sub-pixel precision	28
3.5 Outliers	29
3.6 Validation	30
3.7 Tracer Particles & Seeding	31
4 Heuristic Control	33
4.1 Manual Control	33

4.2	Control of Object	34
4.3	Steady State Control	38
4.4	Mean Velocity Control	44
5	Conclusions & outlook	47
5.1	Future work	48
A	Position control loop	49
	Bibliography	51

Nomenclature

Acronyms

AA4CC Advanced Algorithms for Control and Communications.

ADC Analog-to-digital converter.

API Application Programming Interface.

DAC Digital-to-analog converter.

DC Direct Current.

EMI Electromagnetic Interference.

fps Frames Per Second.

GPIO General-purpose Input Output.

GUI Graphical User Interface.

IC Integrated circuit.

LED Light-emitting diode.

MCU Micro Controller Unit.

MHD Magnetohydrodynamics.

MOSFET Metal-oxide-semiconductor Field-effect Transistor.

PCB Printed circuit board.

PIV Particle Image Velocimetry.

PWM Pulse-width modulation.

SPI Serial Peripheral Interface.

USB Universal Serial Bus.

Mathematical symbols

\mathbb{Z} The set of whole numbers.

Fluid dynamics, the study of fluid movement and interaction with its surroundings, is crucial in a wide range of engineering applications. Magnetohydrodynamics (MHD) is a subbranch of fluid dynamics, focusing on conductive fluids in the presence of magnetic fields. The need for controlling fluids is substantial for applications like fluid pumping and stirring, or for precisely delivering medicine through blood. However, controlling flows still poses significant challenges due to the inherent complexity. In this thesis, I build an experimental platform allowing for the control and measurement of MHD flow. Using this platform, I test various heuristic control strategies to induce and maintain the desired flow. The platform can also be used for testing other control approaches, such as data-driven control.

1.1 Motivation & Goals

There are several existing domains of interest concerning MHD flow. Mostly plasma flow is studied since it is used in plasma generators [Her+06], or in fusion reactors [Gar98]. MHD plays also an important role in astrophysics [PBS11]. The most widespread use of MHD in engineering is liquid metal stirring [Dav16].

This thesis aims to develop an experimental platform for exploring the possibilities of flow control and manipulating floating objects using MHD flow. Additionally, develop and assess the performance of different heuristic control approaches on said platform.

1.2 My Contribution

Due to the extensive scope of this project, the work was divided between myself and Bc. Adam Uchytíl, who is currently completing his master's thesis.

My contribution to this project is the development of two control boards that allow for setting the voltage on the electrodes and controlling the current flowing through them, the implementation of flow measurement, through a method of image recognition of tracer particles dispersed in the fluid, the creation of a velocity profile measurements dataset, and the development of a heuristic strategy for object position control and mean velocity flow control.

The mechanical construction of the platform was done with the help of Adam Uchytíl and Křištof Pučejdl.

Adam also helped me with testing electrodes, electrolytes (see [Section 2.5](#)), and tracer particles (see [Section 3.7](#)).

1.3 Structure

First, in [Section 1.4](#), I introduce the principles of MHD. In [Chapter 2](#), I describe the development of control boards, construction of the platform, and selection of electrodes and electrolytes. In [Chapter 3](#), I present the Particle Image Velocimetry (PIV) algorithm, preprocessing and postprocessing methods, and selection of tracer particles. I then construct a dataset of velocity profile measurements and develop control algorithms in [Chapter 4](#).

1.4 Magneto hydrodynamics

MHD is the study of moving conductive fluids interacting with magnetic fields.

This thesis studies the movement of conductive fluid induced by the *Lorentz force*. Lorentz force \mathbf{F} per unit volume, acting on a conductive fluid is given by

$$\mathbf{F} = \rho_e \mathbf{E} + \mathbf{J} \times \mathbf{B}, \quad (1.1)$$

where \mathbf{E} is an electric field, \mathbf{B} is a magnetic field and ρ_e is charge density. Finally, \mathbf{J} is the current density, derived from Ohm's law it holds that

$$\mathbf{J} = \sigma(\mathbf{E} + \mathbf{u} \times \mathbf{B}), \quad (1.2)$$

where \mathbf{u} is the velocity of the fluid and σ is its conductivity. Note that the first term, in [Equation \(1.1\)](#) is, for speeds much less than the speed of light, negligible [[Dav16](#)].

In our setup which is described in [Chapter 2](#), the electric field \mathbf{E} is induced by potential on electrodes submerged in the conductive fluid. The magnetic field \mathbf{B} is produced by coils under the fluid, perpendicular to \mathbf{E} . Image explaining the principle is depicted in [Figure 1.1](#).

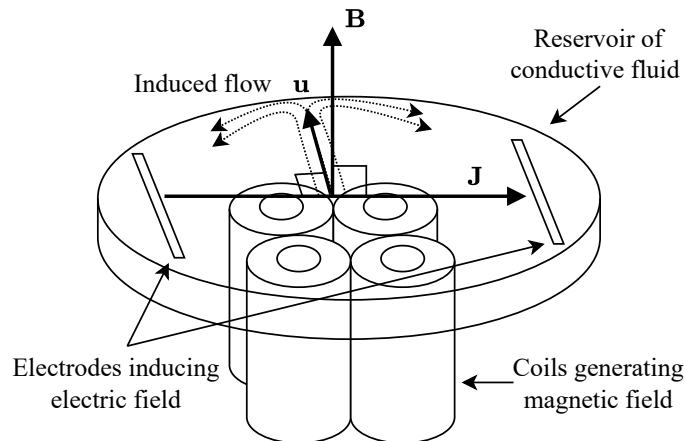


Figure 1.1: Principle of MHD flow being induced by Lorentz force.

One of the most important parameters that describe the MHD flow is the *Magnetic Reynolds number*

$$R_m = \mu \sigma v l, \quad (1.3)$$

Where μ is the permeability of free space, σ is the electrical conductivity of the fluid, l is a characteristic length scale and v is the characteristic speed of the fluid. In [Dav16] it is stated that for $R_m \ll 1$ the velocity \mathbf{u} of the fluid has very little to no effect on the magnetic field \mathbf{B} .

The values used in the presented setup are in these orders of magnitude:

$$\mu \approx 10^{-6} \text{ H/m},$$

$$\sigma \approx 1 \text{ S/m},$$

$$u \approx 10^{-2} \text{ m/s},$$

$$l \approx 10^{-1} \text{ m},$$

Resulting in

$$R_m \approx 10^{-9} \ll 1.$$

Even when using the liquid metals with $\sigma \approx 10^6 \text{ S/m}$, it still holds that $R_m \ll 1$, so the fluid does not affect \mathbf{B} , meaning no magnetic induction happens.

The flow of the fluid is then described by *Navier-Stokes equations*. A comprehensive explanation of the governing equations of MHD can be found in [Dav16].

The setup requires coils to generate magnetic fields, allowing for control over the strength of the magnetic field by adjusting the current flowing through them. Additionally, the ability to apply and modulate voltage on electrodes is necessary for adjusting the electric field. Lastly, a camera is essential for obtaining feedback from the flow. The experimental platform comprises four main components, each described in detail in this chapter. They are also visualized in [Figure 2.1 \(a\)](#).

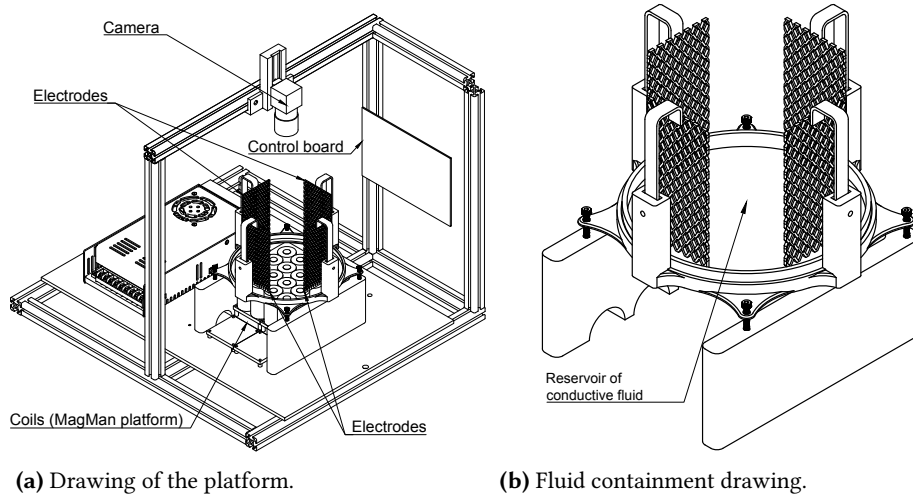


Figure 2.1: Platform overview.

Communication between individual parts is shown in [Figure 2.2](#).

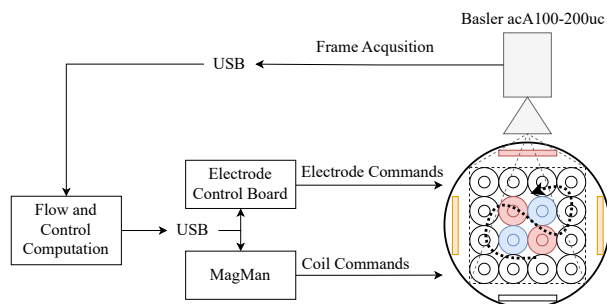


Figure 2.2: Platform communication diagram.

2.1 Camera

The used camera is *Basler acA1300-200uc*¹. Its settings, configured via *pypylon*² a Python Application Programming Interface (API) for interfacing with the camera, are listed in [Table 2.1](#). Other parameters not listed in the table are set to default values specified in the camera's documentation.

The camera is positioned at an exact height to capture a $10\text{ cm} \times 10\text{ cm}$ area depicted in [Figure 2.3](#). The area is a square inscribed into a circular glass dish, which holds a conductive liquid and has a diameter of 147 mm.

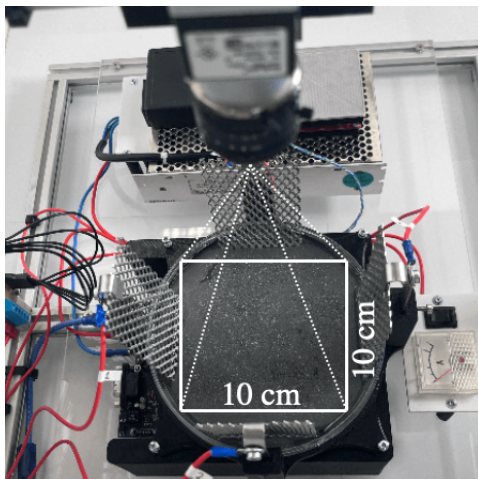


Figure 2.3: A $10\text{ cm} \times 10\text{ cm}$ area captured by the camera.

Table 2.1: Camera setting values used in *pypylon* API.

Parameter	Value
AcquisitionFrameRate	25 fps
Width	1024 px
Height	1024 px
OffsetX	128 px
PixelFormat	Mono8
ExposureTime	10 000.0 μs
Gain	2.0

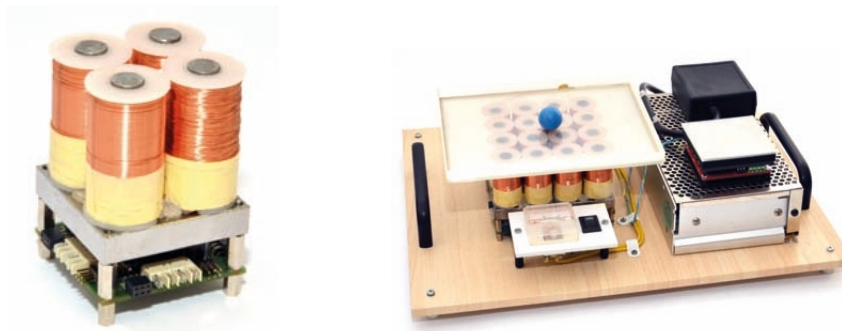
2.2 The MagMan Platform

MagMan is a modular coil platform initially designed for manipulating steel balls using magnetic fields. A single module consists of four coils and is depicted in [Figure 2.4 \(a\)](#). Our platform is built on four of these modules, resulting in a 4×4 grid of coils.

MagMan was developed and constructed by the Advanced Algorithms for Control and Communications (AA4CC) group at the Department of Control Engineering, Faculty of Electrical Engineering. A detailed description of the MagMan platform can be found in [[Zem18](#)].

My work on this platform was to disassemble it from the state that can be seen in [Figure 2.4 \(b\)](#) and create a setup that would suit the needs of MHD better. This includes

- 1 Camera used: <https://docs.baslerweb.com/aca1300-200uc>
- 2 *pypylon* software suite: <https://github.com/basler/pypylon>



(a) Single 4 coils module.

(b) Complete MagMan platform.

Figure 2.4: The MagMan platform (taken from [Zem18], pp. 82)

precise positioning to center the camera above the dish, easy leveling of the dish, and robustness to movement.

With the help of Krištof Pučejdl and Adam Uchytíl, we have incorporated the MagMan into the platform, as can be seen in Figure 2.1 (a), which satisfies all of the specified requirements.

2.3 Voltage Control Electrode Board

The development of this board was primarily the focus of my semestral project. The goal was to construct a board capable of supplying variable voltage to each electrode and measuring the current flowing through each one. This section explains the core functionality and describes the communication API.

2.3.1 Pulse-width modulation

The main functionality of the control board is to supply variable voltage ranging from 0 V to power supply voltage (up to 15 V) to each electrode. This is achieved by switching between 0 V and the power supply voltage using a pulse-width modulated (PWM) signal. The switching is done by a MOSFET driver³, which contains 2 MOSFETs in a half H-bridge configuration along with the necessary switching logic to ensure proper opening and closing of the transistors. The driver can switch a maximum current of 2 A.

The PWM signal is generated by a Light-emitting diode (LED) driver⁴ (moving forward called PWM generator). By adding pull-up resistors to its outputs, the desired functionality is achieved. The driver has 10 individual channels, a resolution of 8 bits, and a PWM frequency of 110 Hz. The PWM duty cycle is set via Serial Peripheral Interface (SPI).

³ MOSFET driver datasheet: <https://cz.mouser.com/datasheet/2/240/media-3323668.pdf>

⁴ PWM Generator datasheet: https://cz.mouser.com/datasheet/2/609/MAX6966_MAX6967-3130958.pdf

Connection between Micro Controller Unit (MCU), PWM generator and MOSFET driver is illustrated in [Figure 2.5](#).

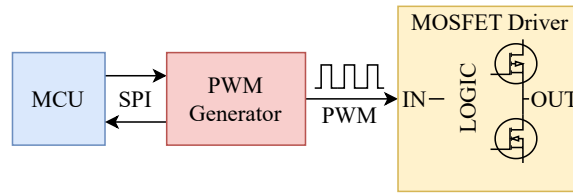


Figure 2.5: Connection between MCU, PWM generator and MOSFET driver.

Correct functionality was verified with the use of an oscilloscope. The measured data, which can be seen in [Figure 2.6](#), show the expected frequency and correct duty cycles.

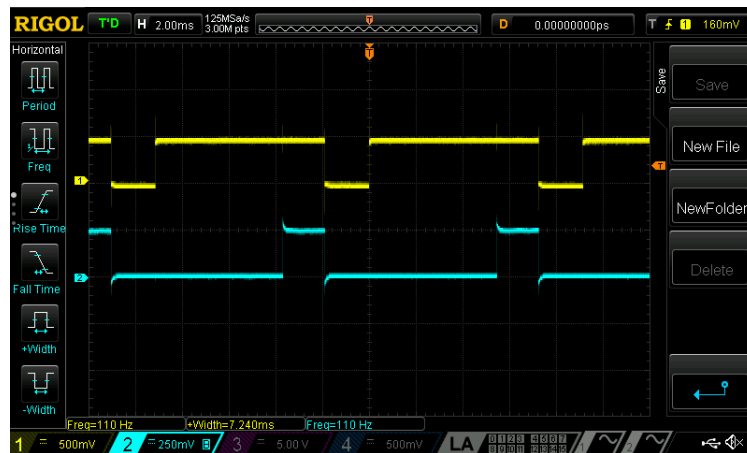


Figure 2.6: Verification of correct switching functionality, yellow signal is 80 % duty cycle, the teal is 20 %.

2.3.2 Current measuring

Although not strictly necessary, the ability to measure the current flowing through each electrode is a useful feature for verifying correct functionality or current-based control. Current is measured by using a so-called shunt resistor on which voltage drop U_R is measured, an example of such circuit is depicted in [Figure 2.7](#). Using Ohm’s law, the current flowing through the resistor is calculated as

$$I = \frac{U_R}{R_{shunt}} \quad (2.1)$$

To not limit the current flow, the resistance of the shunt resistor needs to be in the milli-ohm range. This poses a new challenge in measuring the voltage drop across the

resistor. For example when $R = 10 \text{ m}\Omega$ and $I = 2 \text{ A}$, a voltage of 20 mV needs to be measured. A common approach is to use an operational amplifier to amplify the voltage.

I have selected an *INA240*⁵, ideal for this application due to its enhanced PWM rejection, making it well-suited for measuring current switched by a PWM signal.

A shunt resistor with a resistance of $R = 15 \text{ m}\Omega$ is used. Assuming a maximum current of 2 A flowing through a single electrode, the voltage drop across the resistor is $U_R = 30 \text{ mV}$. Taking into account that the current can flow in both directions, a span of 60 mV needs to be considered. Working with 3.3 V logic levels, the voltage can be amplified by a maximum factor of 55. Therefore, the closest version of the *INA240* amplifier offering a gain of 50 V/V was selected.

The input voltage needs to be offset by $U_{\text{offset}} = 3.3/2 = 1.65 \text{ V}$ since negative voltage requires special handling and most Integrated circuits (ICs) are not designed to withstand it. This means that when no current is flowing a voltage of 1.65 V will be measured on the output. Fortunately, the *INA240* provides two pins, *REF1* and *REF2*, precisely for this purpose. The final schematic can be seen in [Figure 2.7](#). The documentation does not list the values of resistors labeled "R".

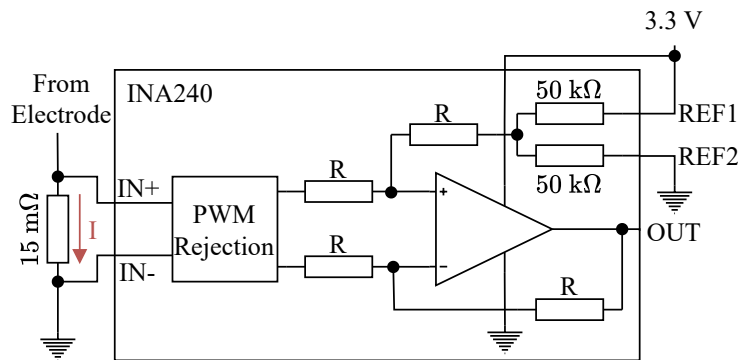


Figure 2.7: Functional diagram of *INA240* and its use to measure the current flowing through a shunt resistor.

Lastly, the output from the amplifier is measured using an Analog-to-digital converter (ADC) expander *MAX11643*⁶ which can measure up to 16 individual channels. Measured voltages can be read out using SPI. Indicated in [Figure 2.8](#) is a connection diagram.

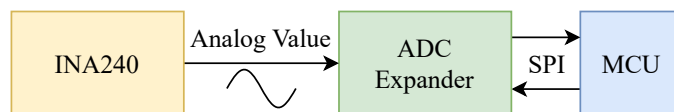


Figure 2.8: Connection between MCU, ADC expander and *INA240*.

⁵ *INA240* datasheet: <https://www.ti.com/lit/ds/symlink/ina240.pdf>

⁶ *MAX11643* datasheet: https://cz.mouser.com/datasheet/2/609/MAX11638_MAX11643-3126997.pdf

2.3.3 Communication API

Communication with the board itself is done with the use of a USB protocol, ensuring the reliable delivery of each message. A C library *libusb*⁷ which implements the protocol is used.

Following is an overview of the types of commands, their format, and the meaning of values, with each value being 1 Byte (8 bits).

Set LED

is a command for setting an on-board LED.

Message Type	Value
0	0 - Off, 1 - On, 255 - Toggle

Set Channels

is for setting channels with indexes "Channel indexes" to duty cycles "Channel duties", where the indexes correspond to numbers written on the Printed circuit board (PCB). To calculate what value to send when needing to set a specific duty cycle (in %), this formula can be used.

$$\text{value} = \text{round}\left(\frac{\text{duty}}{100} \times 255\right) \quad (2.2)$$

Message Type	Number of channels N	Channel indexes	Channel duties
1	1-12	$N \times (1-12)$	$N \times (0-255)$

For example, sending bytes [1, 2, 4, 5, 0, 255] would set electrode 4 to 0 % duty cycle and electrode 5 to 100 % duty cycle.

Set All Channels

has the same functionality as Set Channels without the need to specify channel indexes. Works as if the indexes were ordered 1 → 12.

Message Type	Channel duties
2	$12 \times (0-255)$

⁷ libusb library: <https://libusb.info>

Clear Channels

sets all channels to 0 % duty. Single byte of value 3 needs to be sent.

Measure Current

is a command for measuring currents flowing through given "Channel indexes" and returning the measured data.

Message Type	Number of channels N	Channel indexes
4	1–12	$N \times (1-12)$

The returned data is output from ADC in the same order as were "Channel indexes".

Message Type	Number of channels N	ADC output
7	1–12	$N \times (0-255)$

The current can be calculated as

$$I = \frac{\frac{ADC}{255} 2.5 - 1.65}{50 \cdot 0.015}. \quad (2.3)$$

Measure All Currents

is for measuring current flowing through all channels in consecutive order $1 \rightarrow 12$. A single byte of value 5 needs to be sent. The same message as in "Measure current" is returned.

Get Firmware Version

Returns firmware version of the board in "X.X.XX" format. A single byte with a value of 6 needs to be sent. Returned data is in format

Message Type	Version
8	$(0-255) \cdot (0-255) \cdot (0-255) (0-255)$

2.3.4 Summary

Figure 2.9 illustrates the complete block diagram of the voltage control electrode board. All of the components in the dashed area are repeated twelve times.

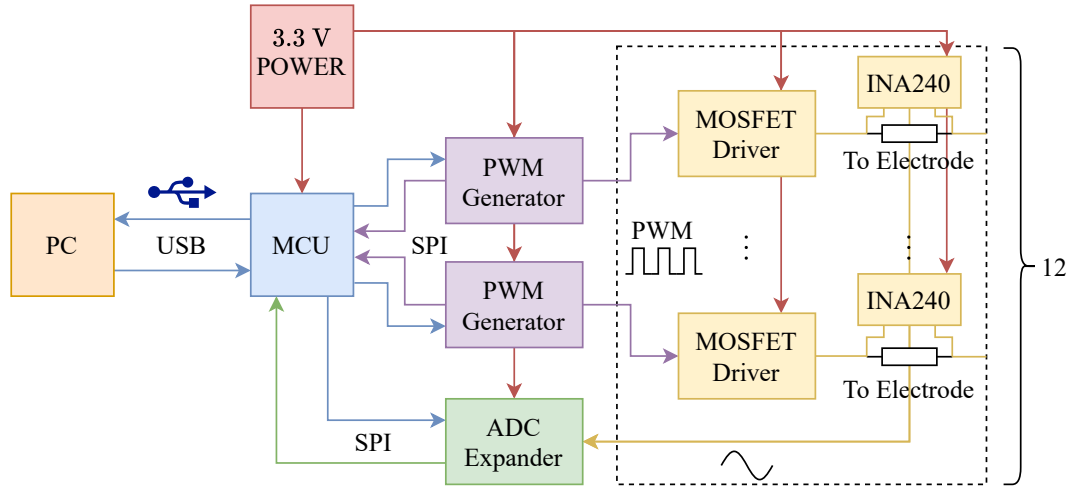


Figure 2.9: Block diagram of voltage control electrode board.

The finished board, depicted in Figure 2.10, required some modifications due to incorrect component footprints and a design flaw that omitted pull-up resistors on the outputs of PWM generators. The full schematic of the board can be found on AA4CC GitLab⁸.

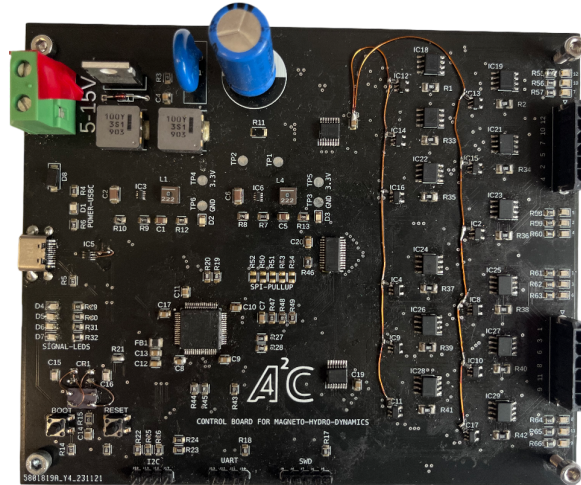


Figure 2.10: Photo of finished voltage control electrode board.

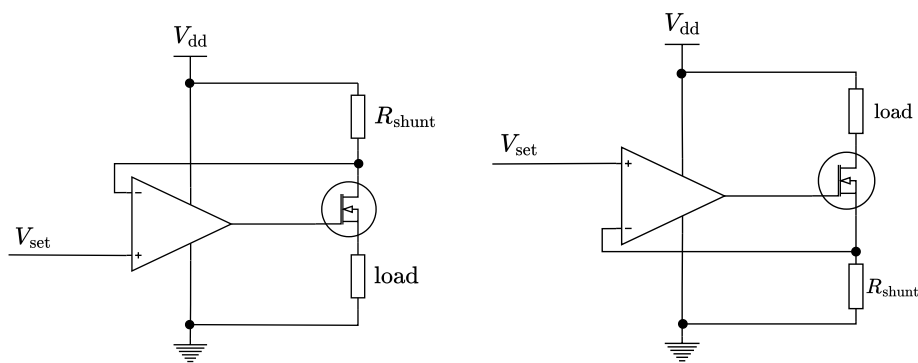
⁸ Schematic: https://gitlab.fel.cvut.cz/aa4cc/mhd/hardware/-/blob/main/voltage_source_v2/SCHEMATIC.pdf

2.4 Current Control Electrode Board

In addition to electrolytes, we want to do experiments with liquid metals. They are six orders of magnitude more conductive than electrolytes used with the voltage electrode board (see Section 2.5). The resistivity of liquid metals is in order of $\text{m}\Omega \text{ m}$ [VZV77], therefore the current would easily surpass the limit of 2 A. Hence new control board was developed. The development process is described in this section.

2.4.1 Voltage-Controlled Constant Current Source/Sink

A voltage-controlled constant current source/sink circuit is needed. This means that while the input voltage remains constant, the output current also remains constant. I was deciding between two designs depicted in Figure 2.11.



(a) Voltage-controlled constant current source design. (b) Voltage-controlled constant current sink design.

Figure 2.11: Voltage-controlled constant current circuits.

- The disadvantage of this design is that setting the current via V_{set} is dependent on V_{dd} . To set current I , voltage $V_{\text{set}} = V_{\text{dd}} - I \cdot R_{\text{shunt}}$ needs to be applied to the positive rail of the operational amplifier. When the V_{dd} exceeds the logic level of 3.3 V of the MCU, an additional amplifier is needed to set the current to 0 A.
- This design solves the issue since to set current I , voltage $V_{\text{set}} = I \cdot R_{\text{shunt}}$ needs to be applied. So I chose this design.

With an additional requirement of being able to either have the electrode as a current sink or have it connected to V_{dd} , some changes had to be made. A half H-bridge configuration was used similarly as in Section 2.3.1. The final design of a single channel is visualized in Figure 2.12.

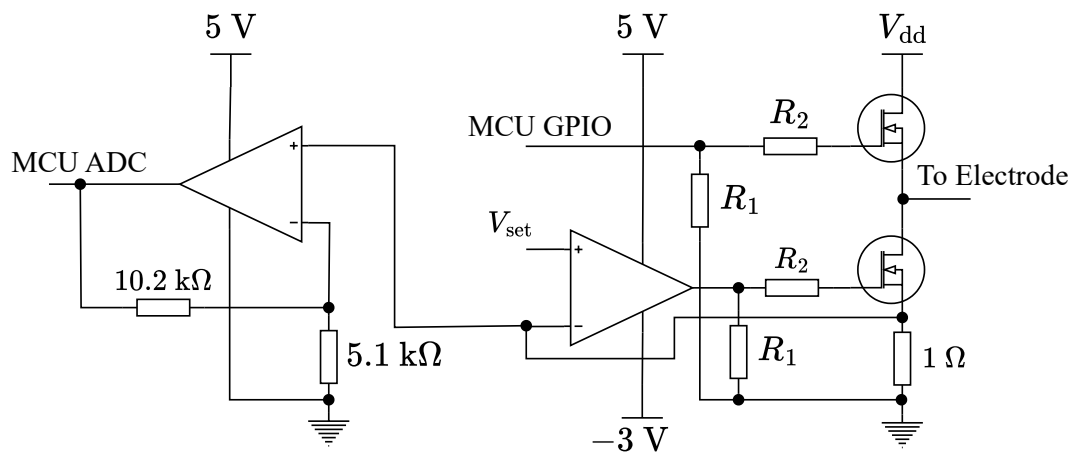


Figure 2.12: Schematic of a single channel.

The shunt resistor is $1\ \Omega$, resulting in a simple conversion where $V_{\text{set}} = 1\ \text{V}$ equals a current of $1\ \text{A}$. The resistors $R_1 = 470\ \text{k}\Omega$ and $R_2 = 100\ \Omega$ are present to ensure a proper discharge of the MOSFETs and to protect MCU General-purpose Input/Output (GPIO) pins.

A negative voltage reference for the amplifier responsible for driving the current-sinking MOSFETs is needed to ensure proper closing of the MOSFETs. How this voltage is provided is described in [Section 2.4.3](#). Lastly, a second operational amplifier, in a non-inverting configuration with voltage gain $A_V = 1 + \frac{10200}{5100} = 3$, was added. This way, the proper functionality of the current sink can be validated without loading it, due to the high input resistance of opamps. The voltage is amplified to utilize the whole range of ADCs.

2.4.2 Low-pass filter

To set the current, analog value V_{SET} is required. However, the selected MCU lacks enough digital-to-analog converters (DAC) to control all twelve channels. As an alternative, I propose utilizing the internal timers of the MCU to generate PWM signals.

The signal cannot be used directly as the V_{SET} because the opamp is sufficiently fast to amplify the signal. Leading to switching high current on high frequency, not satisfying the goal of setting constant current and causing Electromagnetic Interference (EMI). To create an analog signal, a low-pass filter is used. By filtering out higher frequencies of the signal, only the direct current (DC) component remains (in an ideal case).

First-order filter

A first-order low-pass filter can be made by connecting a resistor and a capacitor in series and taking the output from their junction, as shown in [Figure 2.13](#).

When selecting the R and C values, it is important to balance the cut-off frequency $f_c = \frac{1}{2\pi RC}$ (the frequency at which the output signal is $\approx 70\%$ of the input signal amplitude)

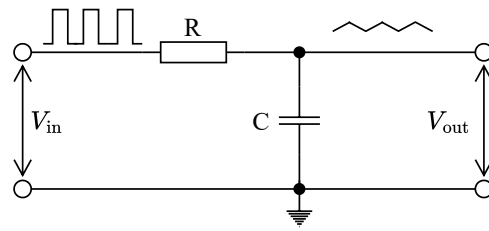


Figure 2.13: First order low-pass RC filter.

and the time constant $\tau = RC$. Lowering the f_c increases the τ , which extends the transient response time. I chose to generate the PWM with a 100 kHz frequency and selected $R = 4000 \Omega$ and $C = 10 \text{ nF}$, resulting in cut-off frequency $f_c \approx 4000 \text{ Hz}$. The *gain* at a frequency of 100 kHz needs to be calculated. Gain is defined as

$$G = \frac{V_{\text{in}}}{V_{\text{out}}}, \quad (2.4)$$

where V_{in} is the input voltage and V_{out} is the output voltage. To simplify the calculation of V_{out} , a harmonic input signal is assumed. Although PWM is not harmonic, it is composed of an infinite series of harmonic signals of different frequencies [GG95], so the result can then be applied to all frequencies individually. The output voltage is calculated as

$$V_{\text{out}}(j\omega) = V_{\text{in}}(j\omega) \frac{X_c}{R + X_c}, \quad (2.5)$$

where X_c is capacitive reactance. By substituting into Equation (2.4) and then plotting the $20 \log G$ at different frequencies, the frequency response, or so-called Bode plot, is created, shown in Figure 2.14.

The bode plot depicts, that at $f = 100 \text{ kHz}$ the signal would be attenuated by a factor of 26. This would not result in sufficient damping of higher frequencies. Therefore, higher damping is necessary. Lowering the cut-off frequency is not plausible, as it would increase the time constant, which is not desired.

Second-order filter

By adding a second RC filter on the output of the first one, a second-order filter is created. When using the same R and C values as in the first-order filter, the output voltage can be calculated as

$$V_{\text{out}}(j\omega) = V_{\text{in}}(j\omega) \frac{Z}{R + Z}, \quad (2.6)$$

where

$$Z = \frac{(R + X_c) \cdot X_c}{R + 2X_c}. \quad (2.7)$$

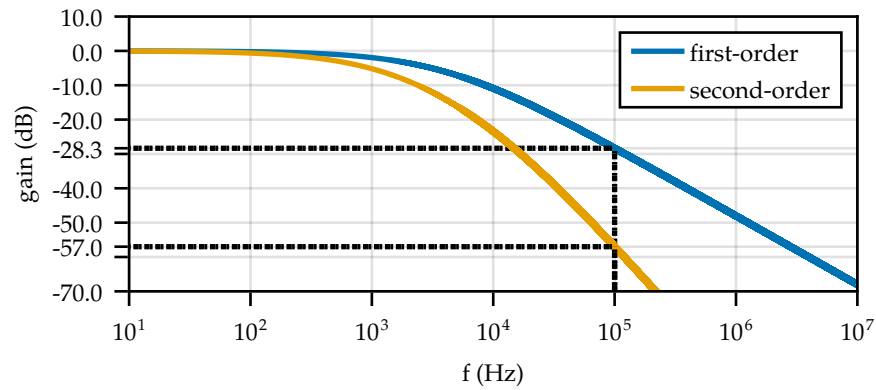


Figure 2.14: Comparison of frequency responses of first and second-order RC filters.

By examining the frequency response and comparing it with the first-order filter, the damping is significantly higher, at ≈ 700 , as depicted in Figure 2.14. To limit the maximum possible current to 1 A, a voltage divider is added to the output of the second filter. Now, a 50 % duty cycle corresponds to 0.5 V on the output of the voltage divider. The resistors used in the divider are $R_1 = 226 \text{ k}\Omega$ and $R_2 = 105 \text{ k}\Omega$, with the output taken across R_2 .

To validate the calculations, I simulated the circuit using *LTspice*⁹ software. Figure 2.15 illustrates the comparison between the simulated data and the measurement of the real circuit.

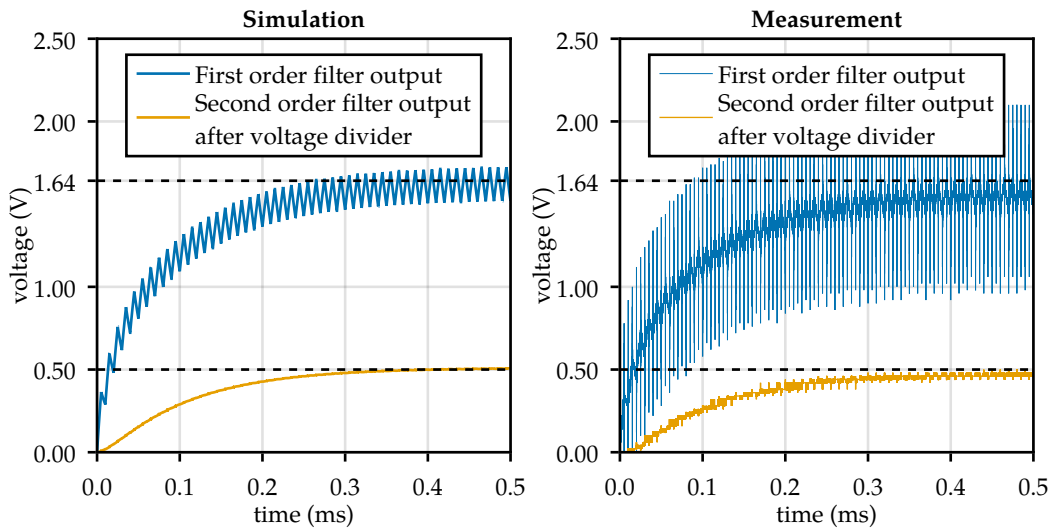


Figure 2.15: Comparison between simulation and measurement of the RC low pass filter with PWM input set to 50 % duty cycle.

⁹ LTspice software: <https://www.analog.com/en/resources/design-tools-and-calculators/ltspice-simulator.html>

2.4.3 Inverting charge pump

To ensure that the current-sinking transistor is fully closed when no current should be flowing, the operational amplifier driving the current-sinking MOSFET requires a negative voltage reference. This is because even rail-to-rail versions cannot operate across the full power supply voltage range. The negative reference is created using an inverting charge pump, its circuit is depicted in Figure 2.16. When the input signal is on (3.3 V), the 47 nF capacitor charges with current flowing through D_1 ¹⁰ (I_1 in Figure 2.16). When the input is off (0 V), the 100 nF capacitor charges to negative voltage supplied by the 47 nF capacitor, with current flowing through D_2 (I_2 in Figure 2.16).

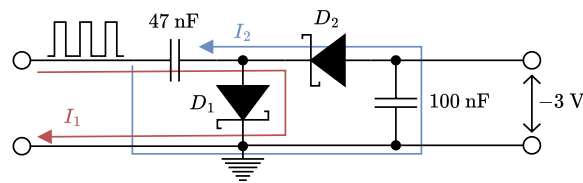


Figure 2.16: Inverting charge pump circuit.

The pump is driven by a PWM signal from the MCU with a frequency of 25 kHz and a 50 % duty cycle. To verify the circuit's functionality, I simulated it. The comparison between simulation and measurement is shown in Figure 2.17. The comparison shows that the real charge pump operates similarly to the simulation when a 350 Ω load is connected to its output. This is to be expected, considering that the output is connected to twelve opamps. Although the output voltage is not ≈ -3 V as in the unloaded charge pump, it is still sufficient to close the MOSFETs.

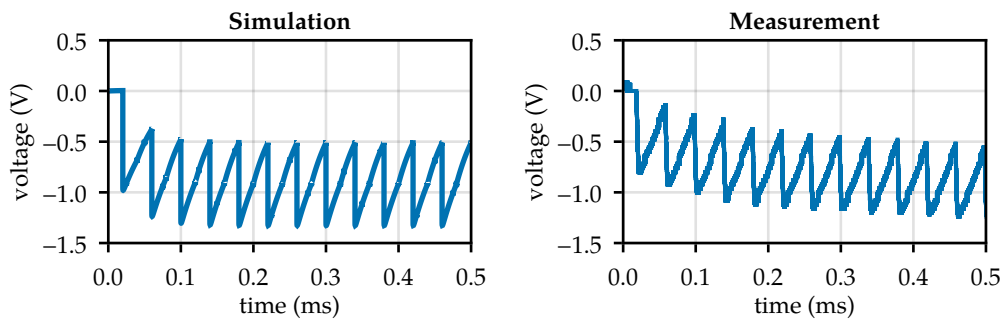


Figure 2.17: Comparison between simulation and measurement of charge pump.

¹⁰ Schottky diodes D_1, D_2 datasheet: https://cz.mouser.com/datasheet/2/115/DIOD_S_A0011683586_1-2543674.pdf

2.4.4 Communication API

Same as in [Section 2.3.3](#) USB communication is used. Again each value is one byte.

Set LED

is the same command as in [Section 2.3.3](#).

Set Channels

is for setting channels with "Channel indexes" to "Channel states", where "Channel states" can be set to values explained in [Table 2.2](#).

Table 2.2: Description of "Channels state" values.

Channel state value	Description
0 – 360	Sets the channel to sink a current of $\frac{value}{360} A$
720	Disconnects the channel from power supply
1080	Connects the channel to power supply

Values bigger than 255 can not fit into a single byte, due to this reason, the channel state values are split into two bytes. How the splitting is done is described in [Listing 2.1](#).

```

1 // Deconstruction into bytes
2 unsigned short channel_state = 360;
3 unsigned char upper = (channel_state >> 8) & 255; // upper = 1
4 unsigned char lower = channel_state & 255; // lower = 104
5 // Construction from bytes
6 unsigned char upper = 1;
7 unsigned char lower = 104;
8 unsigned short channel_state =
9 ((unsigned short)(((unsigned short)upper << 8) | (unsigned short)lower));

```

Listing 2.1: Construction and deconstruction of values.

The message is then constructed as

Message Type	Number of channels N	Channel indexes	Channel duties
1	1–12	$N \times (1-12)$	$N \times (upper, lower)$

2.4.5 Summary

Figure 2.18 illustrates the complete block diagram of the current control electrode board. All of the blocks in the dashed area are repeated twelve times.

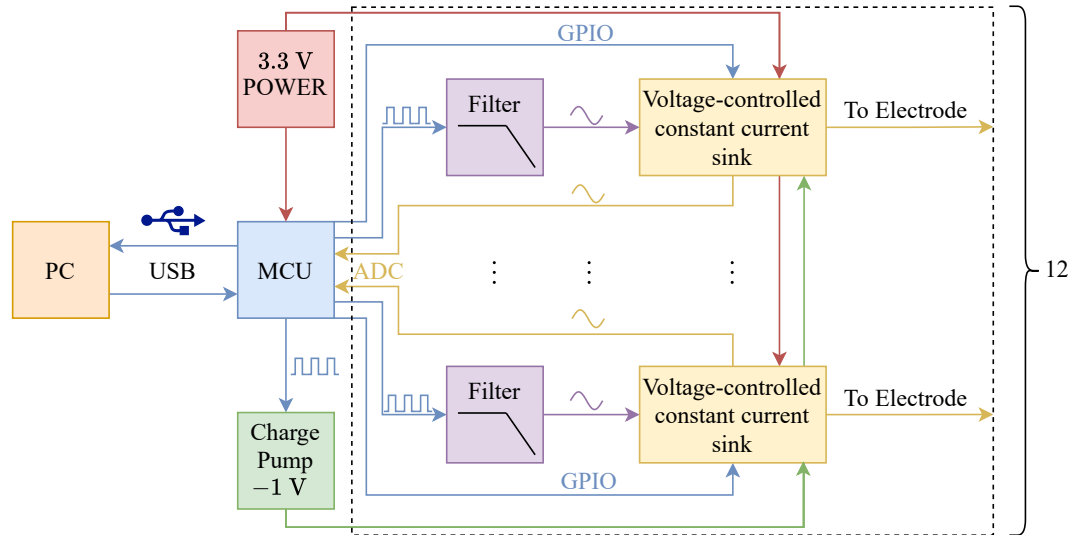


Figure 2.18: Block diagram of current control electrode board.

Figure 2.19 depicts the finished board. This time without any modifications and design flaws. The full schematic of the board can be found on AA4CC GitLab¹¹

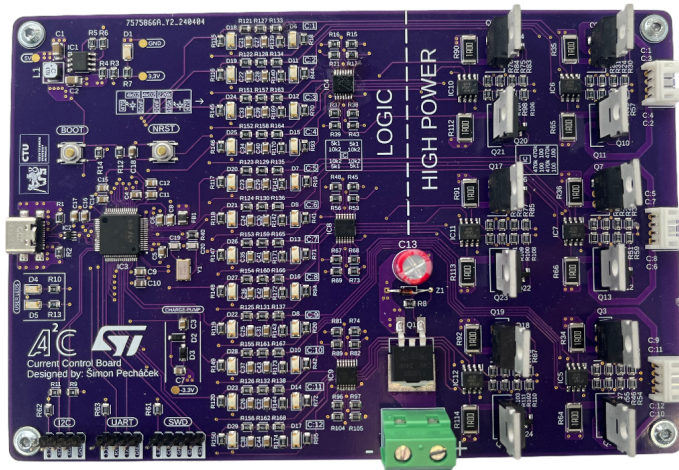


Figure 2.19: Photo of finished current control electrode board.

¹¹ Schematic: https://gitlab.fel.cvut.cz/aa4cc/mhd/hardware/-/blob/main/current_source_v1/Schematic.pdf

2.5 Electrodes & Electrolytes

An electrode is a conductor that connects an electrical circuit to a nonmetallic component, in this case, the control board to an *electrolyte*. Selection of suitable electrodes and electrolytes was challenging, with experiments spanning multiple months. The platform had to be rebuilt multiple times to accommodate the requirements of new electrode and electrolyte combinations. In this section, the tested electrodes and electrolytes, along with their advantages and disadvantages, are described.

2.5.1 Electrodes

1. Gold-plated copper pads

A custom electrode PCB with exposed golden-plated copper pads serving as electrodes. The design is depicted in Figure 2.20.

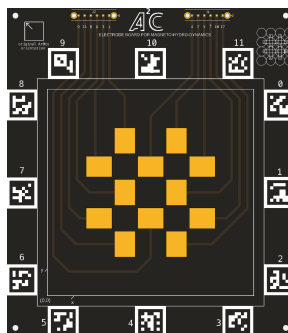


Figure 2.20: Electrode PCB design.

Consisting of twelve electrodes arranged in a symmetrical pattern positioned around the centers of coils. Around the $10\text{ cm} \times 10\text{ cm}$ area, twelve AruCo markers are placed, whose intended use was to convert pixel coordinates into x, y coordinates using homography [Dub09].

2. Platinum-coated razor blades

A standard¹² platinum coated razor blade, although it was not tested whether they were indeed coated with platinum.

3. Platinum-coated titanium electrodes

A $5\text{ cm} \times 10\text{ cm}$ mesh electrode¹³, which should be made from titanium and coated with platinum. The drawing of the electrodes with their arrangement is depicted in Figure 2.1 (b).

¹² ASTRA razor blades used: <https://www.ebay.com/itm/175634476675>

¹³ Titanium electrode used: <https://allegro.cz/nabidka/sitka-titanova-sitka-na-potahovani-sperku-14118956465>

2.5.2 Electrolytes

An electrolyte is a medium containing ions that conduct electricity through the movement of ions, rather than electrons.

1. Na_2SO_4 solution

A 2.5 g of Na_2SO_4 dissolved in 50 mL of deionized water. Resulting in a solution with measured conductivity of $\approx 4 \text{ S m}^{-1}$.

2. H_2SO_4 solution

Associate Professor Pavel Řezanka from VŠCHT recommended low-concentrated sulfuric acid. A 1 mL of 96 % H_2SO_4 in 150 mL of deionized water to be exact. Resulting in a ≈ 0.6 % solution with measured conductivity of $\approx 5 \text{ S m}^{-1}$.

2.5.3 Combinations

1. Custom PCB with Na_2SO_4

This combination is not usable due to the electrolysis of sodium sulfate, releasing oxygen and hydrogen [Liu+23], rendering the measurement of the velocity field unusable. Additionally, the electrodes were slowly corroding, which would require replacing the PCB after a few experiments, something we wanted to avoid.

2. Custom PCB with H_2SO_4

When testing this combination, the golden plating could be wiped off with a cotton swab after ≈ 30 s. After roughly 2 minutes the acid dissolved the copper, disconnecting it from the control board.

3. Razor blades with both electrolytes

The razor blades degraded rapidly. Additionally, solid particles formed, settling at the bottom and causing the electrolyte to turn brown. With the added danger of handling the razor blades, this combination was not selected.

4. Titanium electrodes with Na_2SO_4

The electrodes experienced minimal to no degradation, but similar solid particles formed in the electrolyte, leading to the solution turning brown and introducing errors in flow measurements.

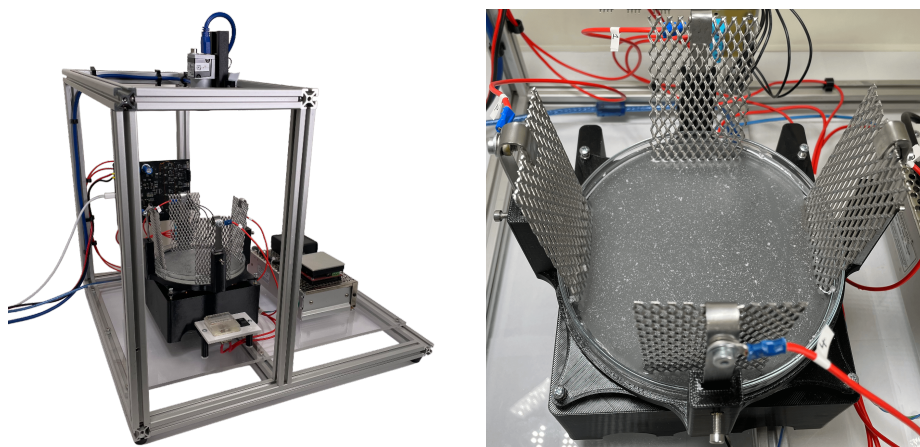
5. Titanium electrodes with H_2SO_4

The electrodes degraded very slowly, and unlike before, the electrolyte did not turn brown. After hours of experimentation, this combination was the best out of all tested.

It is also worth mentioning that the electrolysis was occurring in all combinations, but due to the possibility of positioning the razor blades and titanium electrodes out of the camera frame, this did not pose a problem, as it did with the custom PCB.

2.6 Summary

For the rest of this thesis, the setup is as follows. A 4×4 grid of coils above which a glass dish with a diameter of 148 mm holding electrolyte is positioned. The electrolyte used is 128 mL of an H_2SO_4 solution in which titanium electrodes are partially submerged (to a depth of 8 mm). Electrodes are controlled via a voltage control board with a power supply voltage of 10 V.



(a) Photo of a newly constructed experimental platform. (b) Photo of a glass bowl with sulfuric acid and titanium electrodes.

Figure 2.21: Photos of a newly constructed experimental platform.

Measuring flow in an area with high resolution (more than a few points in space) is a challenging task. Existing methods for achieving high-resolution flow measurements include Particle Tracking Velocimetry (PTV) and Particle Image Velocimetry (PIV).

PIV is an optical used to measure velocity fields in fluids. Unlike other approaches such as pressure probes or anemometers, which may disturb the flow, PIV is non-intrusive. However, as shown in [Section 3.7](#), this is not always the case. The speed of the flow is not measured directly but indirectly with the use of *tracer particles*.

In this chapter, the principle and math behind the algorithm are explained. Employed pre-processing and post-processing techniques are described, followed by validation of the algorithm. Finally, an overview of the tracer particles and their *seeding* techniques is presented.

3.1 Principle

The PIV algorithm can be described in multiple steps shown in [Figure 3.1](#).

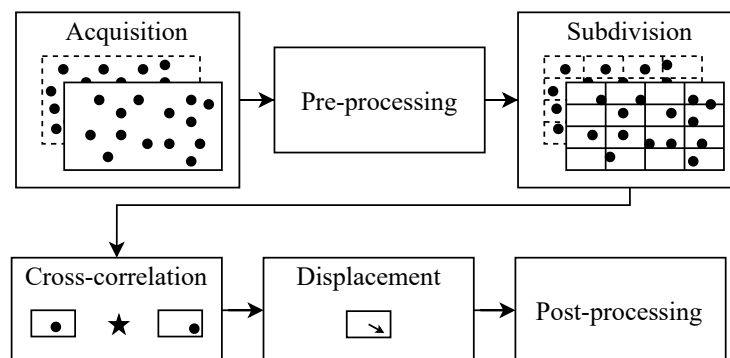


Figure 3.1: PIV algorithm steps.

Acquisition

First, two frames at times t and $t + \Delta t$ are taken. While this may appear straightforward, several factors play a role in the quality of the outcome. Selecting parameters like resolution and framerate affects the selection of parameters in the following steps.

In our application, the camera's maximum resolution, which forms a square shape, was set to 1024×1024 pixels. While a bigger resolution results in finer measurement, the evaluation takes longer and more tracer particles are needed for reliable results.

The framerate was set to 25 fps, since at the mentioned resolution, the algorithm can run comfortably in real time, leaving enough time for other computations.

Pre-processing

The main objective of pre-processing is to enhance particle image contrast and bring particle image intensities to a similar signal level to ensure that all particle images have a similar contribution in the correlation function. This can be achieved through methods such as intensity capping [SLS07] or histogram equalization, as described in [Section 3.3](#).

Subdivision

Here, each frame is subdivided into so-called interrogation windows of size $M \times N$. These windows need to be small enough to ensure that particles move only in linear motion between the two frames (without rotation or deformation), but also large enough that the maximum displacement of particles between two frames is not bigger than that window. With larger windows, it is also easier to have a sufficient number of particle pairs between frames.

The smallest window where these criteria were satisfied was of size 32×32 . This resulted in sufficient vector field resolution. To use windows of dimensions 16×16 , a slightly higher framerate would be necessary, as the displacement of particles between two frames was too large.

Cross-correlation & Displacement

Cross-correlation is computed for every interrogation window. The displacement of particles is then determined from the computed result. This step is described in detail in [Section 3.2](#).

Post-processing

Two post-processing methods are used. First, a sub-pixel precision algorithm, which allows for precise measurements, is described in [Section 3.4](#). Second, an outlier detection algorithm is used to detect and replace outliers, described in [Section 3.5](#).

3.2 Evaluation

To extract the displacement information from two consecutive interrogation windows, cross-correlation needs to be computed. A comprehensive explanation of PIV cross-correlation is given in [KA92].

3.2.1 2D Convolution

First a 2D circular convolution of discrete functions f and h is defined as

$$f(x, y) \circledast h(x, y) = \sum_{m=-a}^a \sum_{n=-b}^b f(m, n)h(x - m, y - n), \quad (3.1)$$

where \circledast denotes circular convolution, $a = \frac{(M-1)}{2}$ and $b = \frac{(N-1)}{2}$ where $M \times N$ denotes the size of the interrogation window in this case.

In this application, $f(x, y) = I(x, y)$ and $h(x, y) = I'(x, y)$ where $I(x, y)$ is the intensity distribution of the first window and $I'(x, y)$ is the intensity distribution of the consecutive window.

Circular convolution is used for periodic functions, but $I(x, y)$ and $I'(x, y)$ are not periodic, leading to unwanted effects such as aliasing or the introduction of high-frequency noise.

3.2.2 Relation between convolution and correlation

For 1D functions, the relation is defined as

$$[f(t) \star h(t)](t) = \left[\overline{f(-t)} * h(t) \right](t), \quad (3.2)$$

where \star denotes correlation, and $\overline{f(-t)}$ is complex conjugate of function $f(-t)$.

Expanding this to 2D functions and assuming only real discrete functions, the cross-correlation is defined as

$$f(x, y) \star h(x, y) = \sum_{m=-a}^a \sum_{n=-b}^b f(m, n)h(x + m, y + n). \quad (3.3)$$

Looking at Equation (3.1), convolution and correlation differ by geometrically rotating function h by 180° [GW07].

3.2.3 Computation in frequency domain

Using the convolution theorem which states

$$\mathcal{F}\{f \star h\} = \overline{\mathcal{F}\{f\}} \cdot \mathcal{F}\{h\}, \quad (3.4)$$

where \mathcal{F} denotes the Fourier transform operator, the cross-correlation can be computed as

$$R_{II'} = \mathcal{F}^{-1} \left\{ \overline{\mathcal{F}\{I\}} \cdot \mathcal{F}\{I'\} \right\}, \quad (3.5)$$

where \mathcal{F}^{-1} is the inverse Fourier transform and $R_{II'}$ denotes cross-correlation of intensity distributions I and I' .

When calculating the Fourier transform, a DFT or FFT [CT65] algorithm is used. The same problem as in spatial correlation arises, these algorithms assume the functions to be periodic. The correlation data is therefore also periodic. If the particle displacement between two windows is bigger than $N/2$ (suppose a window of size $N \times N$), the Nyquist-Shannon sampling theorem has been violated, and the result is aliased [Sha49]. $N/2$ is therefore the lowest maximum displacement threshold, but in [Raf+18], a bigger margin of $N/3$ or even $N/4$ is suggested.

3.2.4 Displacement

Displacement can be determined from cross-correlation by finding argmax.

$$\operatorname{argmax}_{x,y} R_{II'}(x,y) \quad (3.6)$$

3.2.5 Example

The example in Figure 3.2 shows a single 32×32 window pair calculation. These windows are artificially generated by my script and they represent an ideal case, where all particles move in the same direction without violating Nyquist-Shannon sampling theorem. No background noise is present, and lighting conditions do not change between two frames. Lastly, there is a sufficient number of particle pairs present in both frames. The goal is to come as close as possible to these conditions in a real application.

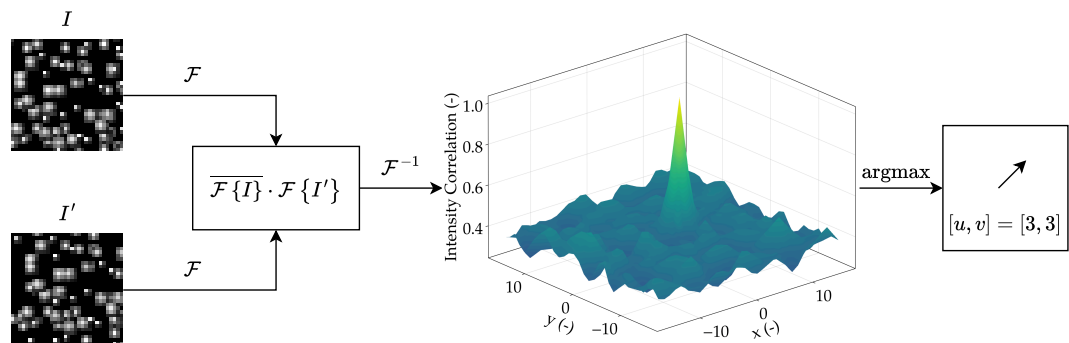


Figure 3.2: Example of single window pair calculation.

3.3 Histogram equalization

Histogram equalization is a pre-processing method used to increase the contrast of images. This is achieved by evenly distributing the densely populated intensity values [GW07].

Assume a grey-scale image \mathbb{A} of L grey levels. Let n_i be the number of pixels with an intensity value of i where $0 \leq i < L$. The probability of an occurrence of a pixel with intensity i is then

$$p(i) = \frac{n_i}{WH}, \quad (3.7)$$

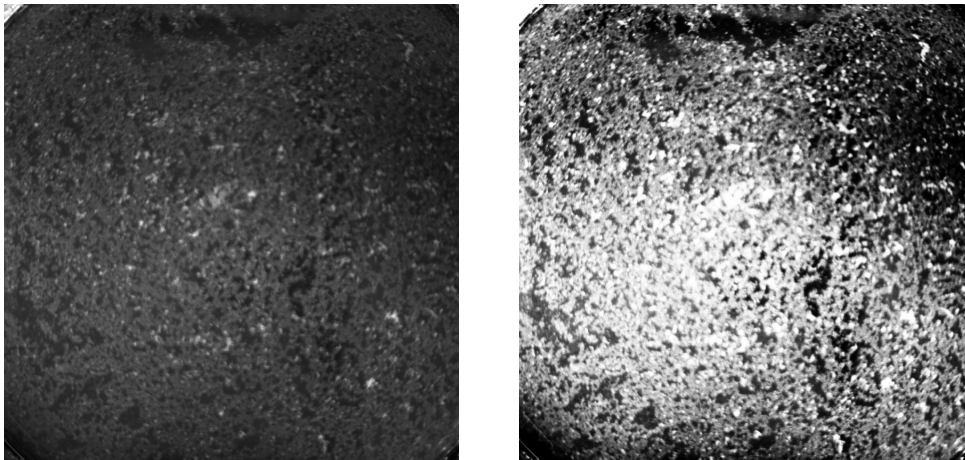
where W is the width of the image and H is its height. The cumulative distribution function is defined as

$$\text{cdf}(i) = \sum_{j=0}^i p(j). \quad (3.8)$$

Now a mapping $T(\cdot)$ needs to be found, such that $\mathbb{B} = T(\mathbb{A})$ where \mathbb{B} is a transformed image with as uniform intensities as possible. The previously defined cumulative distribution function does exactly this.

The mapping of the old intensity value x to the new intensity value of x' is calculated using this formula

$$x' = \text{round}(\text{cdf}(x) \cdot (L - 1)) \quad (3.9)$$



(a) Image before histogram equalization.

(b) Image after histogram equalization.

Figure 3.3: Comparison of image before and after histogram equalization.

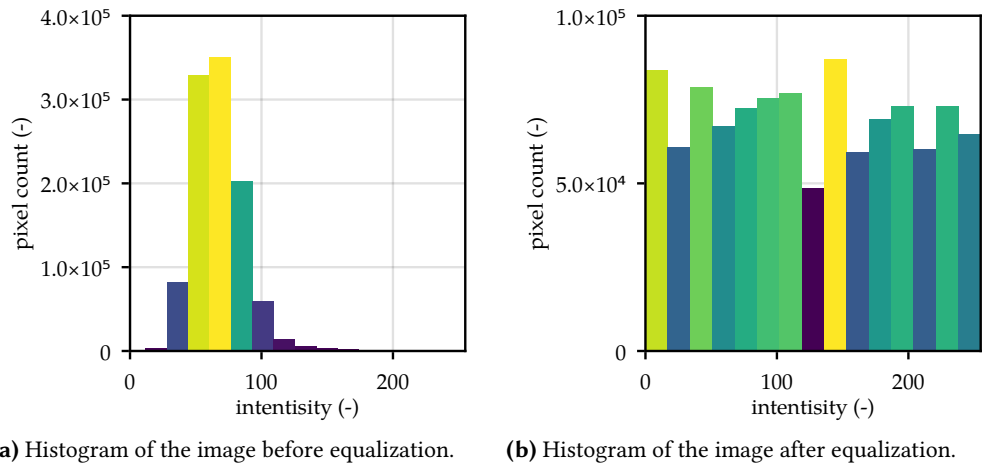


Figure 3.4: Comparison of image histograms before and after histogram equalization.

3.4 Sub-pixel precision

A post-processing method calculated immediately after the displacement determination, as the correlation data is needed for each calculation. Since the input data is discretized, the displacements acquired from the cross-correlation can only be whole numbers $x, y \in \mathbb{Z}$. More information can be extracted from the cross-correlation because, in real applications, not all particles in an interrogation window move the same amount between two frames. A mean particle displacement can be estimated. The most used methods to estimate mean displacement are so-called "three-point estimators". They work by fitting a function to the correlation data. Most used functions are either Gaussian or parabolic.

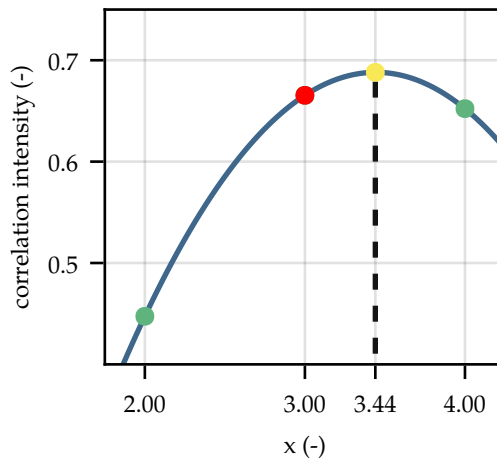


Figure 3.5: The principle of parabolic three-point estimator.

The principle for peak estimation in x direction is shown in Figure 3.5, the same holds for a peak in y direction. The red point is a peak which was acquired by calculating $x_0, y_0 = \operatorname{argmax} R_{II'}$. The green points are directly next to the peak with values $R_{II'}(x_0 - 1, y_0)$ and $R_{II'}(x_0 + 1, y_0)$. The yellow point is the new peak in the x direction estimated by parabolic function. The new peak can be calculated using these formulas taken from [Raf+18].

$$x'_0 = x_0 + \frac{R_{II'}(x_0 - 1, y_0) - R_{II'}(x_0 + 1, y_0)}{2R_{II'}(x_0 - 1, y_0) - 4R_{II'}(x_0, y_0) + 2R_{II'}(x_0 + 1, y_0)} \quad (3.10)$$

$$y'_0 = y_0 + \frac{R_{II'}(x_0, y_0 - 1) - R_{II'}(x_0, y_0 + 1)}{2R_{II'}(x_0, y_0 - 1) - 4R_{II'}(x_0, y_0) + 2R_{II'}(x_0, y_0 + 1)} \quad (3.11)$$

3.5 Outliers

A second post-processing method is used to remove and replace outliers in the vector field data. An outlier is a data point that significantly differs from other observations. In PIV, outliers are computed displacements that do not match with their neighborhood. The neighborhood is a 3×3 or 5×5 area around the outlier. An example of an outlier (u_0, v_0) is shown in Figure 3.6.

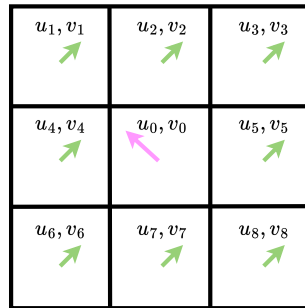


Figure 3.6: Example of an outlier in PIV data.

The algorithm for detecting outliers is taken from [WS05]. A slight modification was made to also process the edges of the measured data, where the entire neighborhood is not available.

Assume a vector field \mathcal{V} of size $2 \times R \times C$, having u and v components being the displacements in x and y direction respectively, still containing outliers. Pseudo-code to detect them is described in Algorithm 1.

The edges were processed by taking smaller neighborhoods. For example, if u_4, v_4 from Figure 3.6 was being tested, the neighborhood would be $\cup = \{u_1, u_2, u_0, u_7, u_6\}$ and the same for \mathbb{V} . The same procedure as described in Algorithm 1 is then used.

Algorithm 1: Outlier detection algorithm

Input: \mathcal{V}

Output: \mathcal{V}'

- 1 Copy \mathcal{V} into \mathcal{V}' ;
 - 2 For each r, c where $1 \leq r < R - 1, 1 \leq c < C - 1$ obtain neighborhood $\mathbb{U} = \{u_1, u_2, \dots, u_8\}$ and $\mathbb{V} = \{v_1, v_2, \dots, v_8\}$, excluding u_0 and v_0 (see Figure 3.6);
 - 3 Calculate median u_m from \mathbb{U} ;
 - 4 Calculate residual $r_i = |u_i - u_m| \forall i = 1, \dots, 8$;
 - 5 Calculate median r_m from $\{r_1, r_2, \dots, r_8\}$;
 - 6 Calculate residual of $u_0, r_0^* = \frac{|u_0 - u_m|}{r_m + \epsilon}$;
 - 7 Repeat from step 3 this time using the \mathbb{V} neighborhood of displacement vectors, resulting in two residuals $r_{0,u}^*$ and $r_{0,v}^*$;
 - 8 If $\sqrt{(r_{0,u}^*)^2 + (r_{0,v}^*)^2} > 2, \mathcal{V}[:, r, c]$ is outlier do $\mathcal{V}'[1, r, c] = \text{mean } \mathbb{U}$ and $\mathcal{V}'[2, r, c] = \text{mean } \mathbb{V}$.
-

3.6 Validation

To test whether the algorithm works as expected, I have written a script that generates two consecutive images of particles, examples of which can be seen in Figure 3.2. The displacement in each interrogation window is known, as the particles are generated using the displacement value. As stated in Section 3.2.5, these images represent an ideal case.

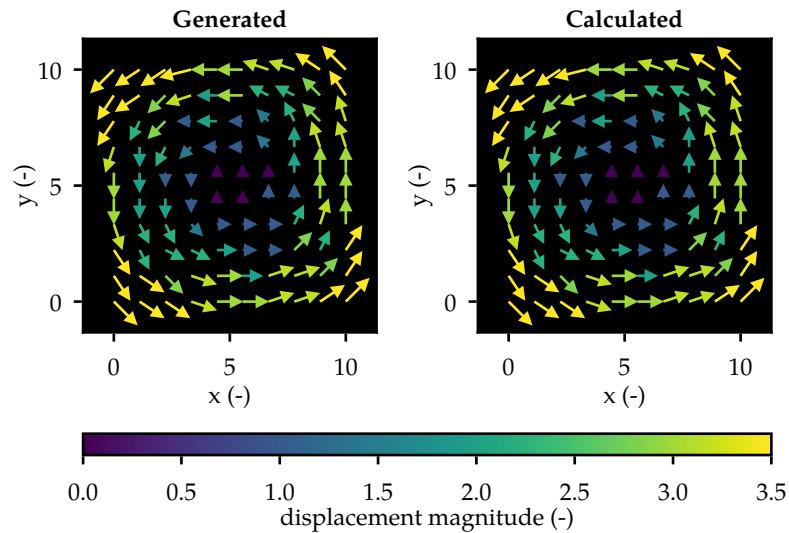


Figure 3.7: Comparison of generated and calculated vector fields.

The comparison in [Figure 3.7](#) is for images 320×320 pixels with interrogation window size 32×32 pixels and maximum displacement of 3 px in each direction. The results show that when providing the algorithm with a sufficient number of particle pairs, consistent lighting conditions, displacements within bounds, etc., the algorithm then calculates the correct displacements.

To show performance on real data, a long exposure shot from a video of a flow and the computed displacements are presented in [Figure 3.8](#).

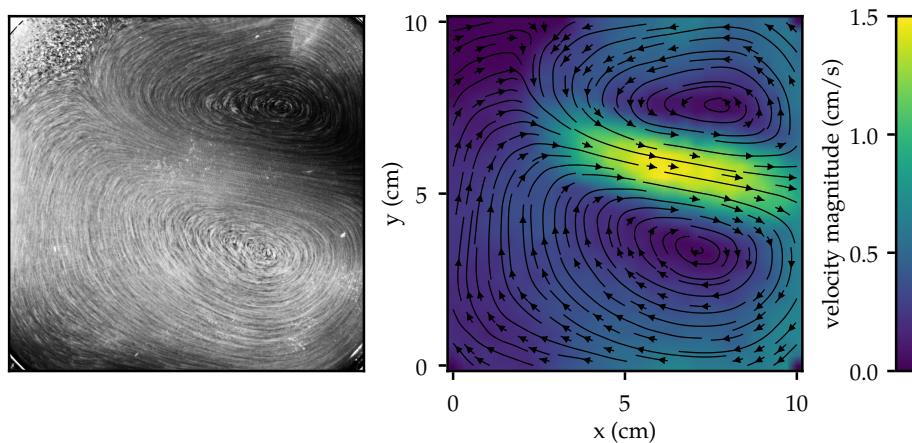


Figure 3.8: Long exposure shot of a flow (left) and computed displacements (right).

3.7 Tracer Particles & Seeding

Tracer particles can range from dispersed oil or air bubbles to glass or plastic spheres. They are supposed to follow the flow faithfully without disturbing it. The types of tracer particles tested are described in this section.

I would like to thank the Institute of Hydrodynamics AS CR and Ing. Jan Haidl, Ph.D. for providing us with the tracer particles.

3.7.1 Tracer Particles

1. Poly methyl methacrylate

Also known as acrylic, with a maximum particle size of $600 \mu\text{m}$. These particles provide reliable measurements, by uniformly and sufficiently covering the whole measured area.

The issue with these particles is their impact on the measured flow. It appears that as they settle, they form an almost solid layer on the fluid's surface. Consequently, the dragging and collisions of the particles affect the flow, example of the effect is depicted in [Figure 3.9](#).

This behavior could be measured and demonstrated by first stirring the fluid with particles (and waiting for the motion caused by the stirring to stop) before measurement, hence disrupting the formation of particles. Then, comparing it with a second measurement without stirring and with a longer wait time. In both cases, the same configuration of coils and electrodes was set, but the measured data is rather different, as visualized in [Figure 3.9](#).

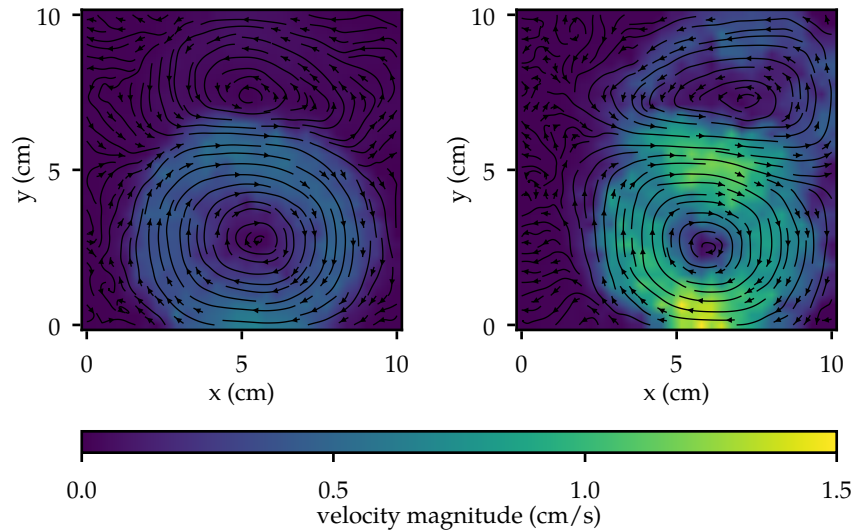


Figure 3.9: Example of different measurements with the same input, where the left was unstirred and the right was stirred before the experiment.

2. ABS

A Polymer with particle sizes ranging from 200 μm to 400 μm , slightly less dense than acrylic. While ABS did not affect the flow, a new issue emerged. The particles tend to form clusters, appearing as one big particle in the video. The clustering creates void areas where no particles are present, rendering displacement calculations impossible.

3.7.2 Seeding

The most significant performance difference, aside from the material used, comes from the method of incorporating (*seeding*) particles into the liquid. Two methods were tried, firstly, dispersing them on the surface and then stirring. Secondly, adding them to the container holding the liquid and shaking it before pouring. When the container is shaken, the particles are no longer held by surface tension, making them sink much faster.

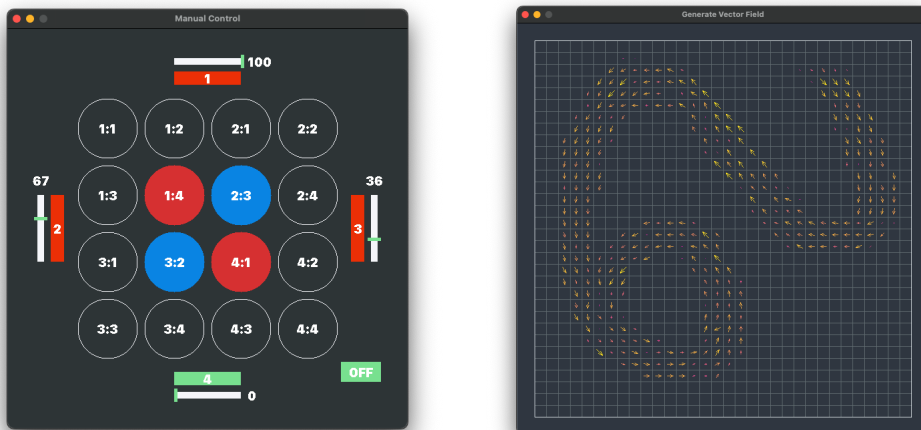
In the end, the acrylic particles were picked, and the dispersing method with thorough stirring was found to perform the best.

In this chapter, heuristic control of MHD flow and objects is explored. First, manual control via a Graphical User Interface (GUI) is shown. Then, an algorithm to manipulate an object is proposed, and finally, control of steady-state flow and its mean velocity is presented.

4.1 Manual Control

For manually controlling the flow, I have developed a GUI in the Julia¹⁴ programming language using the SimpleDirectMediaLayer¹⁵ package.

It allows setting the duty cycle of electrodes and the polarity of coils. This proved to be a useful tool for finding the *alphabet* in Section 4.2 and creating the dataset in Section 4.3. The GUI is depicted in Figure 4.1 (a)



(a) A GUI for manual control of flow.

(b) A GUI for generating reference vector field.

Figure 4.1: Graphical user interfaces developed for easier use of the platform.

The GUI in Figure 4.1 (b) allows for the creation of a vector field by drawing it with a mouse. It is used in Section 4.3 and Section 4.4 for generating desired flows.

¹⁴ Julia programming language: <https://julialang.org>

¹⁵ SimpleDirectMediaLayer package: <https://github.com/JuliaMultimedia/SimpleDirectMediaLayer.jl>

4.2 Control of Object

To manipulate the object freely, it is necessary to generate a flow in any direction at a specific time and place. This is most likely not achievable with heuristic control, hence, the goal is simplified. Instead of moving the object in any direction, the goal is to move the object along the paths shown in Figure 4.2. These paths were chosen based on symmetries within the system (see Section 4.3.1) and experiments conducted with the platform, observing which flows are possible to generate. Through these experiments, I was able to construct a list, or so-called *alphabet*, of electrode and coil input combinations that induce flow in specific positions and directions.

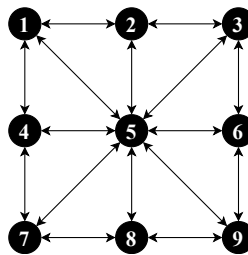


Figure 4.2: Illustration of 9 stationary points and possible translational moves between them.

The controlled object is a circle made from extruded polystyrene with a diameter of 10 mm. Added on top of it is an *AruCo* marker used for position tracking.

4.2.1 Alphabet

The alphabet consists of 16 flows. The first 12 cover directions and positions connecting points 1, 3, 7, 9 in all ways depicted in Figure 4.2. The remaining 4 connects points 2, 8 and 4, 6. In this context, "connecting" means that the paths between the points can be traversed with the specified flow.

Three main directions are sufficient to construct the alphabet, the remaining flows are constructed by exploiting symmetries present in the setup. Symmetries are discussed and verified in Section 4.3.1.

Diagonal Flow

The diagonal flow that connects points 1 and 3, can be induced by coil and electrode configuration presented in Figure 4.3 (a). The induced flow is shown in Figure 4.3 (b). The rectangles represent electrodes. Red means 100 % duty, orange 50 % and white means 0 %. The circles represent the coils, red means maximum current flow through it and blue means reverse polarity of the coil. The dashed square in Figure 4.3 (a) is the 10 cm × 10 cm measured area.

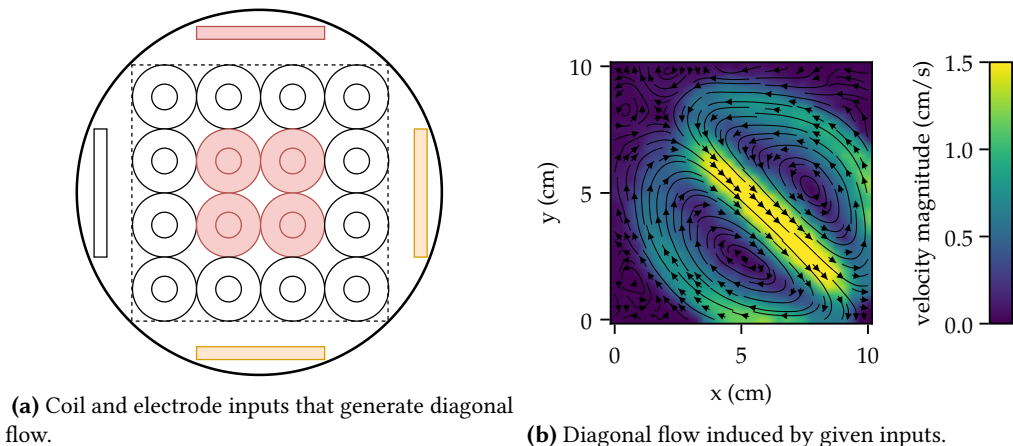


Figure 4.3: Control inputs and the diagonal flow induced by them.

Straight Flow

The flow that connects points 1 and 3 can be generated by a configuration presented in [Figure 4.4 \(a\)](#). The induced flow is shown in [Figure 4.4 \(b\)](#).

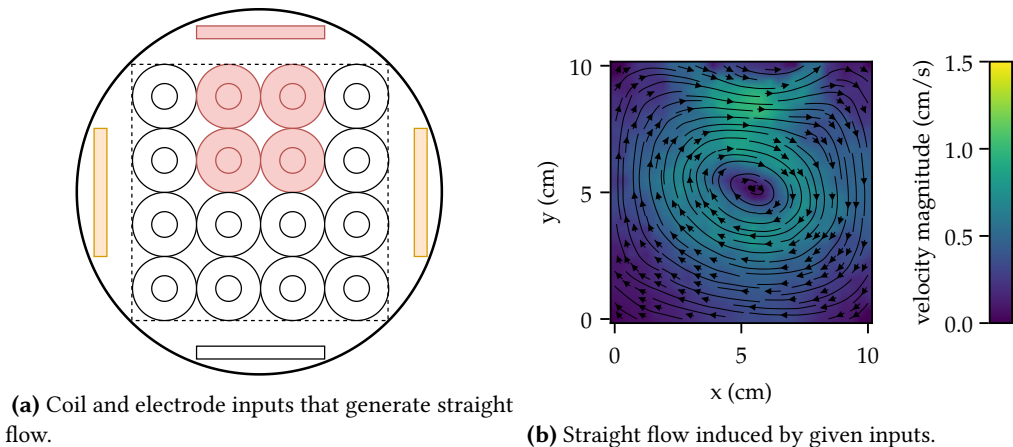


Figure 4.4: Control inputs and the flow induced by them.

The third main move is the direct path through the middle, connecting points 4 and 6, and can be generated by taking coils from [Figure 4.3 \(a\)](#) and electrodes from [Figure 4.4 \(a\)](#).

These flows by themselves are not sufficient to move the object between the points, they are more of a direction in which the object is moved. To stop at the midpoint, for example on point 6 between points 3 and 9, a controller is needed.

4.2.2 Controller

The controller is based on the idea that the amplitude of force, and therefore the speed of the fluid, can be controlled by adjusting the voltage on the electrodes. If the voltages on all electrodes are scaled proportionally, the shape of the flow does not change, but the speed can be controlled.

To get feedback into the controller, the position of the object is measured using *AruCo* markers and *OpenCV* library¹⁶.

To move the object from point $p_i = (x_i, y_i)$ to point $p_j = (x_j, y_j)$, the object needs to be placed close to point p_i at the start. The control loop is described in [Algorithm 2](#). The algorithm with specific values and calculations can be found in the appendix (see [Algorithm 3](#)). The implementation in Julia can be found on AA4CC GitLab¹⁷. The \mathbf{c} are coil inputs (ranging from -1.0 to 1.0 , setting the coil voltage), and \mathbf{e} are electrode inputs (ranging from 0 – 255 see [Section 2.3.3](#)). Finally, $p = (x, y)$ is the position of the object.

Algorithm 2: Position control loop

Input: starting point, goal point, object position

Output: electrode inputs \mathbf{e} , coil inputs \mathbf{c}

- 1 Within the alphabet, find the electrode and coil inputs that connect the starting and goal points;
 - 2 Calculate the distance between the start and the goal (d_{traj});
 - 3 Calculate the distance between the object position and the goal (d_{goal});
 - 4 Calculate the speed of the object;
 - 5 If the d_{goal} is greater than $1/4$ of the d_{traj} , scale the amplitude of electrodes proportionally with the d_{goal} ;
 - 6 If the d_{goal} is less than $1/4$ of the d_{traj} , and the object still has speed, start breaking proportionally with the object speed;
 - 7 Else the goal point is reached;
-

When the destination point p_j is reached a 3 s pause is taken for the flows to diminish. After the pause, the p_j becomes the new p_i and a new destination is picked from a sequence of points. As can be seen in [Algorithm 3](#) a breaking is employed, which also helps with keeping the object in the correct position.

The position of the object can be controlled only along the direction given by points p_i and p_j . If some other force is applied that displaces the object out of the flow, the position can not be corrected. The reason is that even if the needed flows were present in the alphabet, the sudden change would create complex flows. Ideal conditions for the controller to work are proximity to the starting point and no initial velocity.

¹⁶ OpenCV and AruCo markers: https://docs.opencv.org/4.x/d5/dae/tutorial_aruco_detection.html

¹⁷ PIV, GUIs and control algorithms: <https://gitlab.fel.cvut.cz/aa4cc/mhd/julia/control-template>

4.2.3 Results

Presented in Figure 4.5 are the results achieved by using the described alphabet and controller.

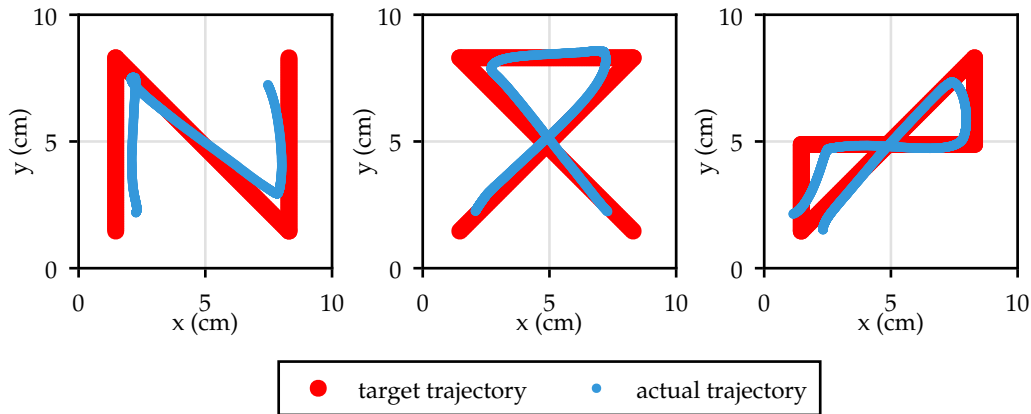


Figure 4.5: Object control results, showing 3 target trajectories (red) and 3 achieved trajectories (blue) using alphabet and controller.

The achieved trajectory on the left is curved toward the center because the straight move presented in Figure 4.4 (b) is not ideal and starts to curve on the sides. The trajectory on the right is incomplete due to surface tension pulling the object toward the dish. This caused most of the problems and explains why the actual trajectories keep their distance from the borders, as the force from the surface tension (at distance ≈ 2 cm from the bowl) is stronger than the induced flow.

In Figure 4.6 are images of object position at different times, corresponding to experiment from Figure 4.5 on the left.

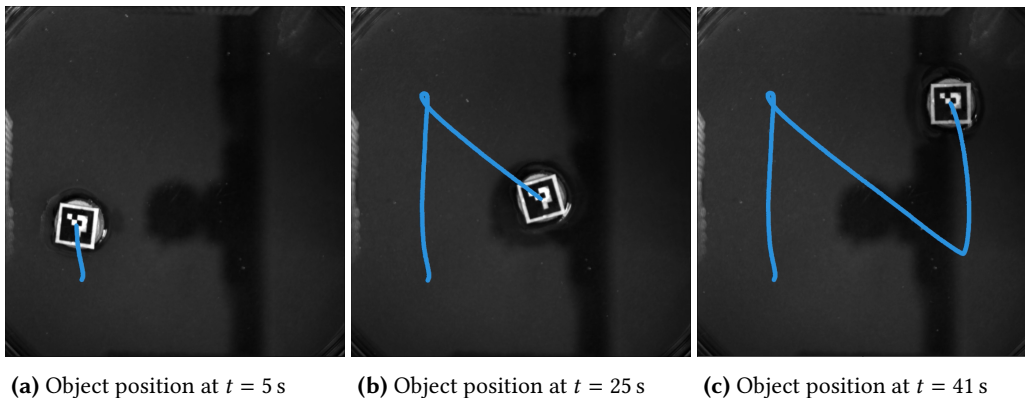


Figure 4.6: Position of the object at different times.

4.3 Steady State Control

Instead of controlling objects to follow simple trajectories, I wanted to generate complex flow shapes (for example, depicted in Figure 4.7), simplifying this task to only deal with steady states. Precise flow control would require computation of MHD equations or other more complex approaches, this is a goal of Bc. Adam Uchytíl's Master's thesis.

I have chosen a different approach, which is based on creating a dataset (similarly as in Section 4.2.1) of coil and electrode inputs and the flow induced by them. The approach differs in that it does not actively control the electrodes and coils. Instead, the control input is calculated at the beginning and then remains constant, without any feedback.

A considerable time is needed for the fluid to stabilize in some cases. If the speed is not zero at the beginning, it can take up to 60 s–80 s to reach a steady state. The second problem is the presence of *limit cycle* oscillations. A limit cycle is a trajectory for which the energy of the system is constant over a cycle [DB09]. They cause problems because the oscillations never stop, hence not reaching a steady state.

The first problem is solved by measuring the flow for 80 s and using the last vector field measurement. The second problem is more challenging due to difficulties in detecting the limit cycles. Due to problems with particles described in Section 3.7, the fluid needs to be manually stirred after each measurement, preventing automatic dataset creation. Hence, the limit cycles were detected and removed from the dataset manually. The drawback is that the dataset is much smaller in size.

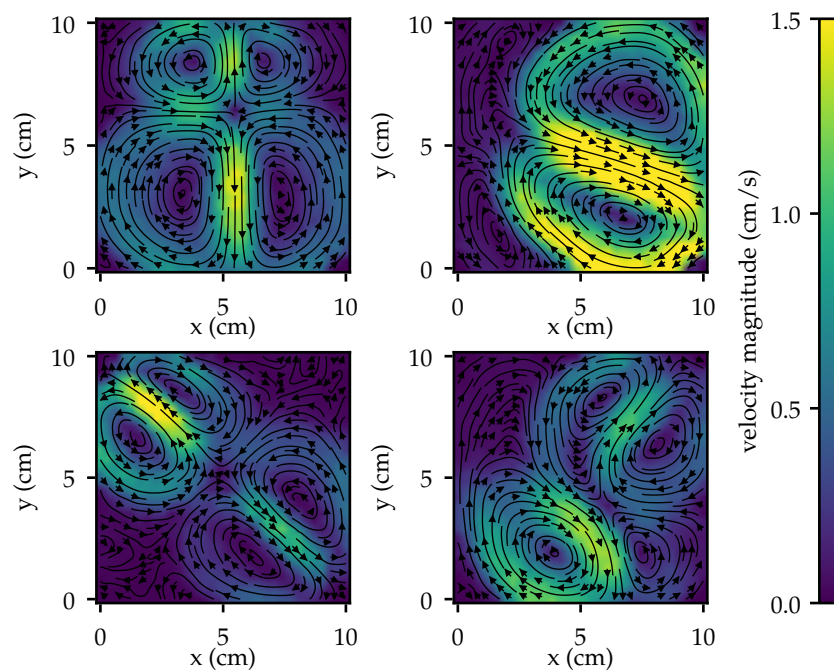


Figure 4.7: Examples of measured steady-state flows.

I have managed to measure 57 different inputs and their corresponding steady states, examples of which are shown in [Figure 4.7](#), with 3 having limit cycles and being removed from the dataset.

Fortunately, there are symmetries in the system that can be exploited, expanding the dataset by a factor of 8. The symmetries are 90° rotation in either direction and mirroring around x and y axis. Only one axis can be used for mirroring if both are used, with rotations by 90° , 180° , and 270° (repeated application of the 90° rotation symmetry), duplicate data is created. Using the 90° , 180° , and 270° rotations and mirroring after every rotation around the x axis, the dataset expansion is achieved.

4.3.1 Validation of Symmetries

To verify the presence of symmetries in the system, I validate them with the presented experiments. First, the initial configuration is set (left in [Figure 4.8](#) and [Figure 4.10](#)) and the flow is measured. Then, the transformation is applied to the inputs and the flow is measured again. The first measured field is rotated/mirrored and compared with the second measurement (labeled "Measured"). The modified fields are labeled "Calculated" in [Figure 4.9](#) and [Figure 4.11](#).

Rotational Symmetry

The rotational symmetry transformation is visualized in [Figure 4.8](#).

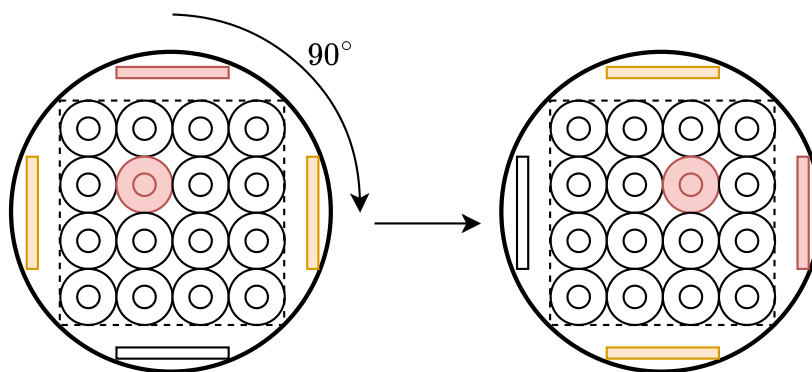


Figure 4.8: Rotational symmetry transformation of inputs.

As can be seen in [Figure 4.9](#), the flow ended up in the same place with a slightly different shape and amplitude, probably caused by the problem with particles described in [Section 3.7](#).

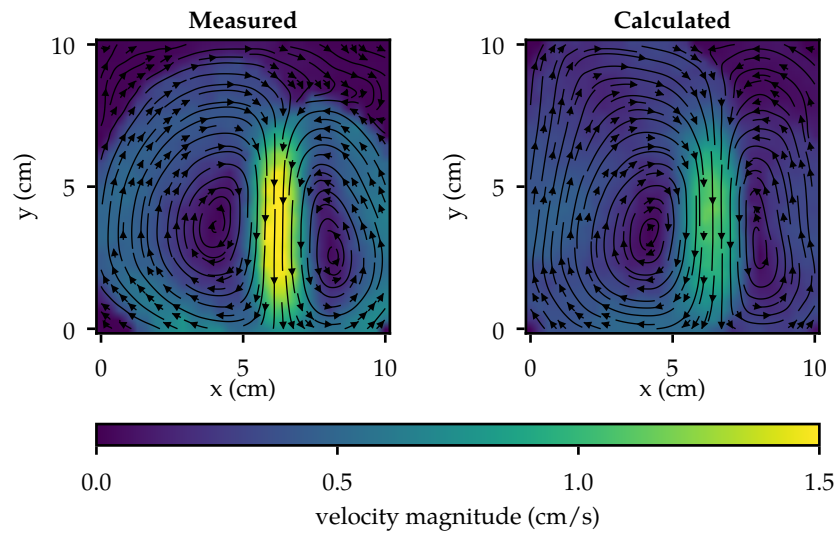


Figure 4.9: Rotational symmetry comparison of calculated and measured data.

Mirror Symmetry

When mirroring the inputs, the polarity of coils needs to be inverted, as visualized in Figure 4.10. The amplitudes of the orange electrodes would also have to be flipped with each other if they had different values.

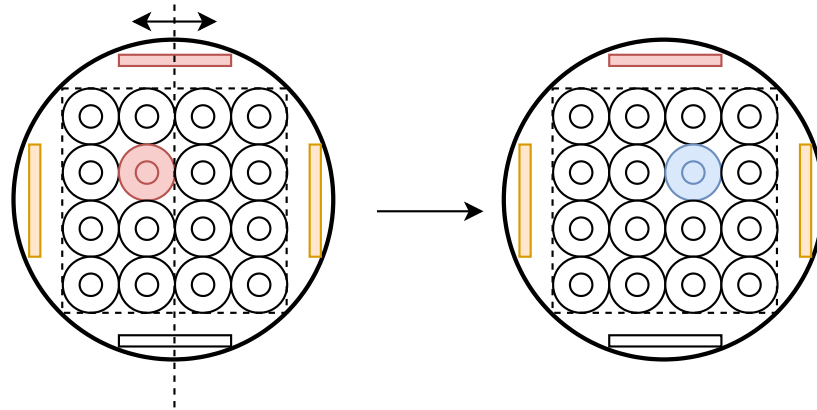


Figure 4.10: Mirror symmetry transformation of inputs.

The Figure 4.11 shows that the mirror symmetry around the x axis is also present in the system. Similarly around y axis.

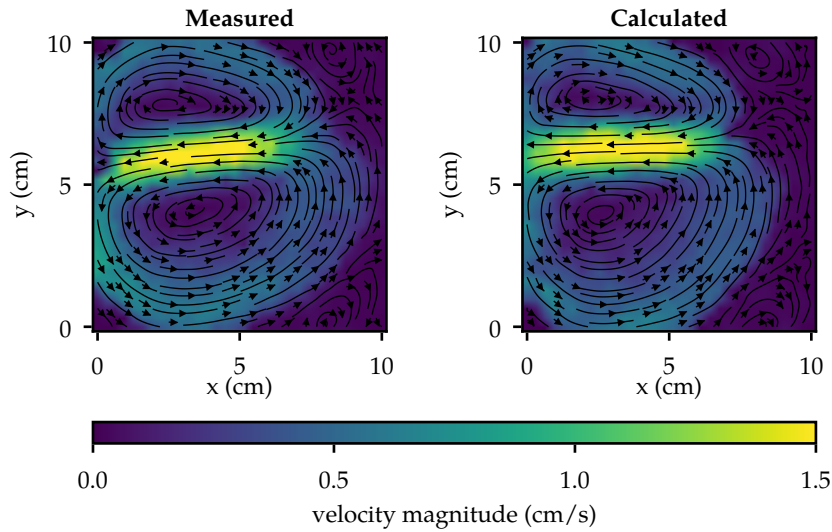


Figure 4.11: Mirror symmetry comparison of calculated and measured data.

4.3.2 Closest Vector Field

It is needed to find the *closest* or most *similar* vector field in the dataset. There are many ways and metrics to define what close or similar means in the context of vector fields.

This is a topic of ongoing research. For example, in [DX08], a statistical algorithm is presented. However, it cannot be used in this situation because the algorithm is invariant to any rotation and shifts. This is not desired, as the measured field cannot be easily generated in a new place or with different rotations (except for the previously mentioned symmetries). There are also algorithms based on locating so-called critical points [Hei+03], but these algorithms cannot be used because critical points are not always present in some types of flows [DX08]. Therefore, I propose metrics for measuring similarities of vector fields.

Minkowski distance

defined as

$$D_p(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^n |x_i - y_i|^p \right)^{1/p}, \quad (4.1)$$

where $\mathbf{x} = (x_1, x_2, \dots, x_n)$ and $\mathbf{y} = (y_1, y_2, \dots, y_n)$ are vectors and n , in this case, is 2.

For $p = 1$ and $p = 2$, the distances are also known as Manhattan and Euclidean, respectively, and they are used in this section. The distance d between two vector fields \mathcal{V}_1 and \mathcal{V}_2 of sizes $H \times W$ containing vectors of size n is defined as

$$d(\mathcal{V}_1, \mathcal{V}_2) = \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} D_p(\mathcal{V}_1(i, j), \mathcal{V}_2(i, j)). \quad (4.2)$$

Suppose \mathcal{V}_1 is the desired field to be found (renaming it to \mathcal{V}_{des}) in a dataset of m vector fields. The closest vector field is found as

$$\operatorname{argmin}_{\{\mathcal{V}_i | i=1, \dots, m\}} d(\mathcal{V}_{\text{des}}, \mathcal{V}_i). \quad (4.3)$$

Cosine Similarity

Having two n dimensional vectors \mathbf{x} and \mathbf{y} (same as in [Section 4.3.2](#)) the cosine similarity S between them is defined as

$$S(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \|\mathbf{y}\|} = \frac{\sum_{i=1}^n x_i y_i}{\sqrt{\sum_{i=1}^n x_i^2} \sqrt{\sum_{i=1}^n y_i^2}}. \quad (4.4)$$

Ensuring that no division by 0 happens, the fields are flattened into nHW long vector, in this case, $n = 2$, $H = 32$ and $W = 32$. Because no vector field is formed solely by zeros, division by zero will not happen. The most similar field to \mathcal{V}_{des} is found as

$$\operatorname{argmax}_{\{\mathcal{V}_i | i=1, \dots, m\}} S(\mathcal{V}_{\text{des}}, \mathcal{V}_i). \quad (4.5)$$

4.3.3 Results

The most similar flows found in the dataset by different metrics are presented in [Figure 4.12](#). Note that the results obtained using the Minkowski distance are closer in amplitude because the fields are not normalized, unlike the fields in cosine similarity.

To help with the search for the most similar shape, *masking* the dataset is proposed. Masking is done by zeroing out the values in areas where the *desired flow* (top left in [Figure 4.12](#)) is 0 in amplitude.

If Euclidean and Manhattan distances are intended to find the closest shape regardless of amplitude, the fields must be normalized. However, normalizing vectors individually is not possible because vectors that are mostly noise would then contribute equally to the search as vectors that are not noise. Therefore, cosine similarity should be used for finding the closest shapes regardless of amplitude.

Out of the 3 metrics presented, the Euclidean distance performed the best, hence is used in [Section 4.4](#).

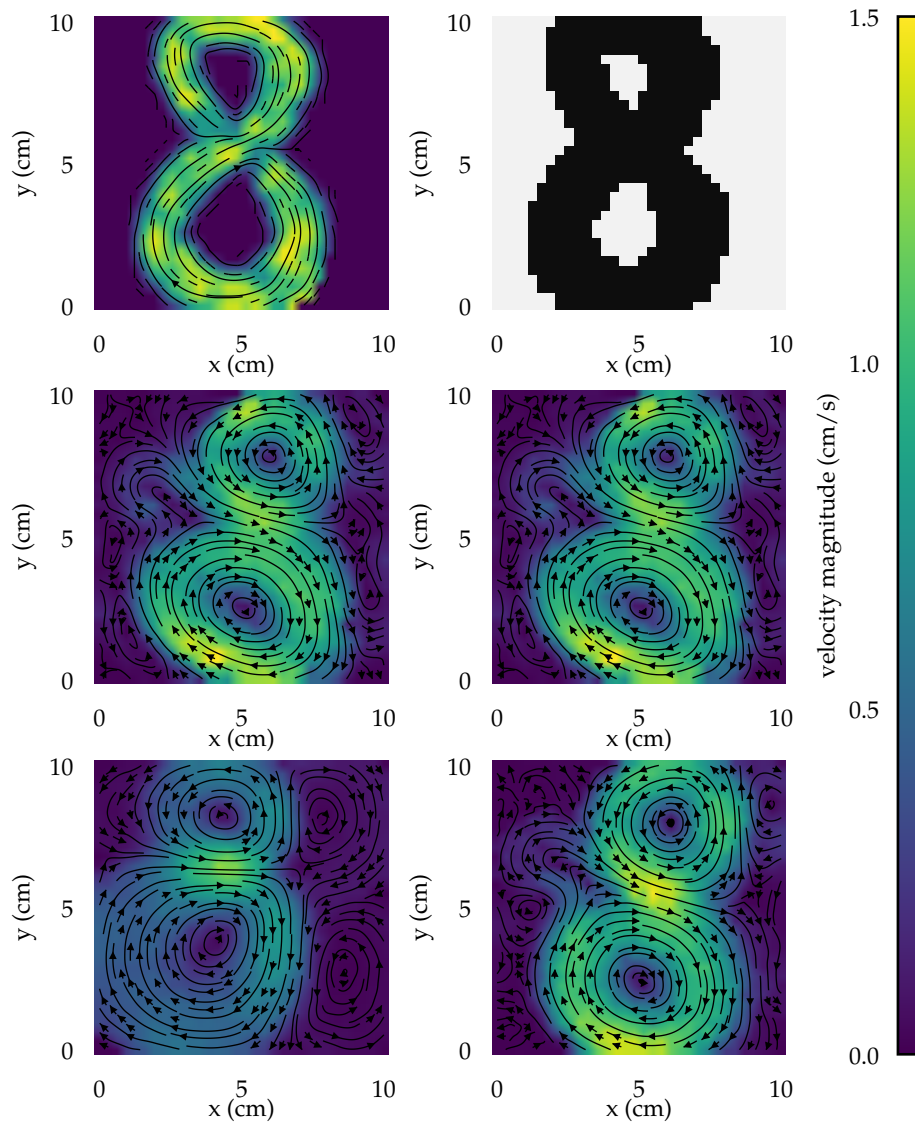


Figure 4.12: Closest vector fields to desired flow with used mask (top row), found by different metrics (middle row and bottom left) and verification measurement of a result found by Euclidean distance (bottom right).

4.4 Mean Velocity Control

Using the same principle as in Section 4.2, the velocity can be varied by scaling the amplitude on all electrodes proportionally. This time not controlling the velocity of an object, but the mean velocity of a desired flow in a specified area. The mean is calculated from vectors selected by a mask, an example of which is shown in Figure 4.12.

The goal is to first find a similar vector field to the desired flow using Euclidean distance as in Section 4.3.2, then to specify target mean velocity, and with the use of a controller to reach and hold that velocity. The maximum velocity that can be reached is calculated from the most similar vector field as most of the flows in the dataset were generated using 100% duty on at least one of the electrodes and higher velocity can not therefore be achieved.

4.4.1 Controller

A controller depicted in Figure 4.13 is used. The e are the electrode inputs that induced the most similar vector field. Additionally, when the *error* from the target is negative, meaning the actual velocity is higher, the coil inputs are inverted, creating flow in the opposite direction, slowing down the fluid. The error is then taken as an absolute value. The constants were chosen based on experimentation and the final values used are $K_p = 10$ and $K_i = 1$. It was observed that the control loop does not have to run as fast as the velocity field measurement, so while the measurement runs at 25 Hz the control loop runs at 2 Hz.

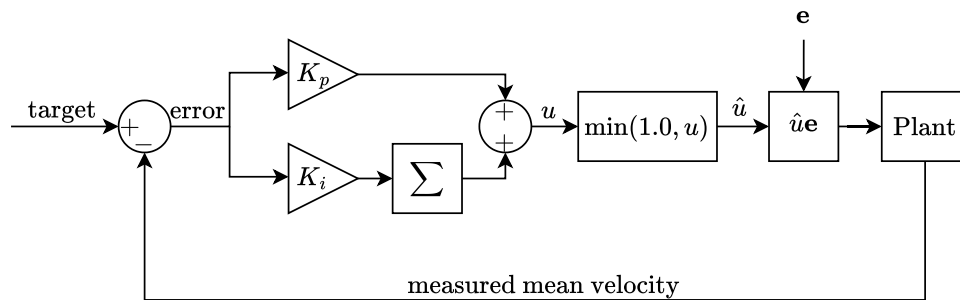


Figure 4.13: Controller used for mean velocity control.

4.4.2 Results

In Figure 4.14, the achieved results of controlling the mean velocity of the *8 shape* from Figure 4.12 are presented. The measured velocity oscillates around the target. This can be explained by the controller not being tuned precisely, but also the whole flow pattern can shift during the experiment, moving in and out of the masked area and changing the measurements. The last target velocity took longer to reach than the previous two because it was the maximum possible mean velocity, thus reaching maximum inputs, with the combined effect of moving out of the masked area.

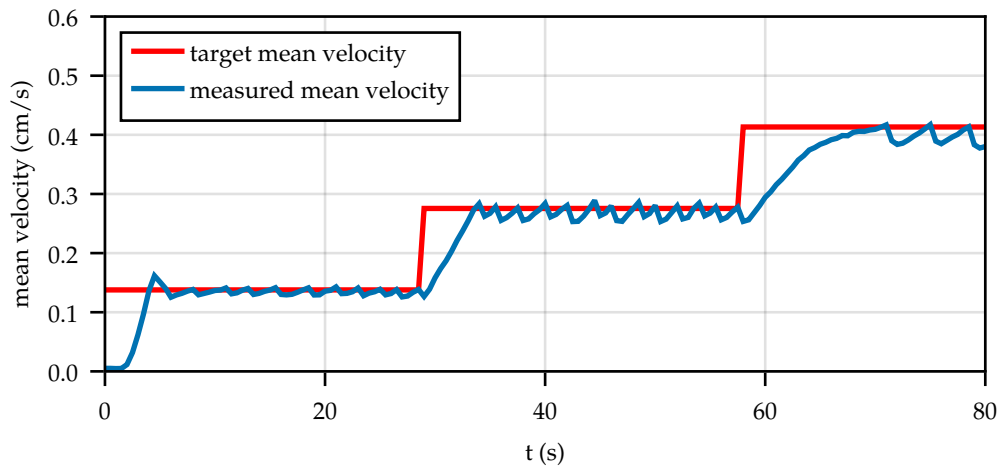


Figure 4.14: Result of mean velocity control.

Presented in Figure 4.15 are also the vector fields at different times from this experiment.

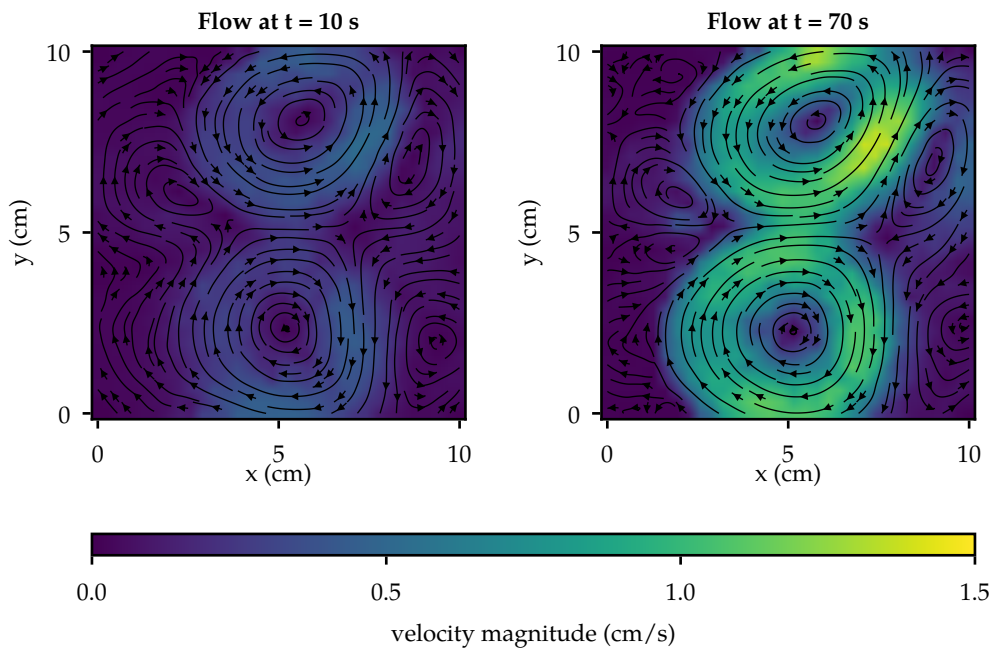


Figure 4.15: Flows at different times during the mean velocity control experiment.

5

Conclusions & outlook

The first goal of this thesis was to construct an experimental platform and design two control boards. The first control board, used for setting the voltage on the electrodes, had some design flaws that could be fixed without the need for a new version. The board works reliably, processing messages at 30 Hz. It can be used for up to twelve channels, switch 2 A of current, and operates at voltages up to 15 V. However, it has a slight limitation with its low PWM resolution of 8 bits.

Initial testing with liquid metals of the second board, designed for setting constant current flowing through the electrodes, showed that the current control works as designed. It can be used with up to twelve channels, allowing a maximum current of 1 A. To control liquid metals, the platform would need to be modified, hence the board was not fully utilized for its intended purpose, but can be used in the future.

The electrode & electrolyte combination showing the best results is the titanium electrodes with the H_2SO_4 solution. The choice is not optimal due to electrode corrosion and slight electrolyte degradation. Adam Uchytíl and I performed multiple experiments on the platform without any failures of the boards or measurements, proving its robustness and reliability.

The second goal of this thesis was to develop a measurement of flows. The algorithm implemented in Julia achieves computations to run at frequency > 100 Hz (on M1 MacBook, with images of size 1024×1024 pixels and interrogation windows of size 32×32 pixels). This performance is way faster than that of pyPIV, an open-source Python package for PIV calculations. The calculated displacements correspond to simulated data, further demonstrating the accuracy of the measurements (see Adam Uchytíl's Master thesis).

The best-performing particles are the Poly methyl methacrylate particles, seeded by dispersing them on the surface followed by stirring. However, they affect the flow in an undesired way.

The final goal of this thesis was to develop heuristic algorithms. Despite their simplicity, the proposed algorithms showed promising results. The object control algorithm successfully manipulates the object along the given paths. It is limited by its inability to correct for external effects and is restricted to movement between only nine fixed points.

The steady-state control is limited to a small dataset but still achieves satisfactory results in finding similar vector fields. Lastly, the mean velocity control is prone to measurement errors and the controller is not precisely tuned, resulting in oscillation around the target. Despite this, the target mean velocities are reached.

5.1 Future work

There is still room for improvement in the selection of electrodes, electrolytes and tracer particles. In the future, I would like to expand the dataset and look more into algorithms for comparing vector fields. I would also like to use the developed current control board and use it for controlling the liquid metals. This will require a partial rebuild of the setup, adding a heating element to keep the metals in a liquid state and a new way of connecting the control board to the liquid metals.

The platform can serve as a foundation for exploring novel data-driven approaches, as the complexity of MHD is a good benchmark for such algorithms. Moreover, it can be used for demonstrating the (in)efficiency of already established algorithms.

A

Position control loop

Algorithm 3: Position control loop

Input: starting point p_i , destination point p_j , object position p

Output: \mathbf{e} , \mathbf{c}

- 1 In alphabet find \mathbf{e} and \mathbf{c} which connect point p_i and p_j ;
 - 2 Calculate distance d between p_i and p_j where $d = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$;
 - 3 Calculate distance d_{goal} between p and p_j where $d_{\text{goal}} = \sqrt{(x - x_j)^2 + (y - y_j)^2}$;
 - 4 Calculate the speed of the object $v = (d_{\text{goal}} - d'_{\text{goal}})/dt$ where d'_{goal} is the previous distance d_{goal} and dt is time elapsed between measurements;
 - 5 If $d_{\text{goal}} > d/4$ then $\mathbf{e} = \mathbf{e}(d_{\text{goal}} - d/4)/d$;
 - 6 If $d_{\text{goal}} \leq d/4$ and $v > 0.1$ cm/s find \mathbf{e} and \mathbf{c} that connect point p_j and p_i , do $\mathbf{e} = \mathbf{e} \min(1.0, v/0.3)$;
 - 7 Else goal reached $\mathbf{e} = \mathbf{0}$ $\mathbf{c} = \mathbf{0}$;
 - 8 $d'_{\text{goal}} = d_{\text{goal}}$;
-

Bibliography

- [CT65] J. W. Cooley and J. W. Tukey. **An Algorithm for the Machine Calculation of Complex Fourier Series**. *Mathematics of Computation* 19:90 (1965), 297–301. ISSN: 0025-5718. DOI: [10.2307/2003354](https://doi.org/10.2307/2003354) (see page 26).
- [Dav16] P. A. Davidson. **Introduction to Magnetohydrodynamics**. Cambridge University Press, Dec. 22, 2016. 575 pp. ISBN: 978-1-107-16016-3 (see pages 1–3).
- [DB09] D. Dutta and J. K. Bhattacharjee. “Limit Cycle Oscillations.” In: *Vibration Problems ICOVP-2007*. Vol. 126. Dordrecht: Springer Netherlands, 2009, 125–135. ISBN: 978-1-4020-9091-2 978-1-4020-9100-1. DOI: [10.1007/978-1-4020-9100-1_14](https://doi.org/10.1007/978-1-4020-9100-1_14) (see page 38).
- [Dub09] E. Dubrofsky. **Homography estimation**. *Master thesis*. Vancouver: University of British Columbia 5 (2009) (see page 20).
- [DX08] H. Q. Dinh and L. Xu. “Measuring the Similarity of Vector Fields Using Global Distributions.” In: *Structural, Syntactic, and Statistical Pattern Recognition*. Vol. 5342. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, 187–196. ISBN: 978-3-540-89688-3 978-3-540-89689-0. DOI: [10.1007/978-3-540-89689-0_23](https://doi.org/10.1007/978-3-540-89689-0_23) (see page 41).
- [Gar98] P. R. Garabedian. **Magnetohydrodynamic stability of fusion plasmas**. *Communications on Pure and Applied Mathematics* 51:9 (1998), 1019–1033. ISSN: 1097-0312. DOI: [10.1002/\(SICI\)1097-0312\(199809/10\)51:9/10<1019::AID-CPA4>3.0.CO;2-G](https://doi.org/10.1002/(SICI)1097-0312(199809/10)51:9/10<1019::AID-CPA4>3.0.CO;2-G) (see page 1).
- [GG95] R. M. Gray and J. W. Goodman. **Fourier transforms: an introduction for engineers**. The Kluwer international series in engineering and computer science SECS 322. New York: Springer Science+Business Media, 1995. 361 pp. ISBN: 978-1-4613-6001-8 (see page 15).
- [GW07] R. C. Gonzalez and R. E. Woods. **Digital image processing**. 3. ed. Upper Saddle River, NJ: Pearson/Prentice Hall, 2007. 954 pp. ISBN: 978-0-13-168728-8 (see pages 25, 27).
- [Hei+03] E. Heiberg et al. **Three-dimensional flow characterization using vector pattern matching**. *IEEE Transactions on Visualization and Computer Graphics* 9:3 (July 2003), 313–319. ISSN: 1077-2626. DOI: [10.1109/TVCG.2003.1207439](https://doi.org/10.1109/TVCG.2003.1207439) (see page 41).
- [Her+06] G. Herdrich et al. **MHD flow control for plasma technology applications**. *Vacuum*. The 5th International Symposium on Applied Plasma Science, 26-30 September 2005, Hawaii, USA 80:11 (Sept. 7, 2006), 1167–1173. ISSN: 0042-207X. DOI: [10.1016/j.vacuum.2006.01.043](https://doi.org/10.1016/j.vacuum.2006.01.043) (see page 1).
- [KA92] R. D. Keane and R. J. Adrian. **Theory of cross-correlation analysis of PIV images**. *Applied Scientific Research* 49:3 (July 1, 1992), 191–215. ISSN: 1573-1987. DOI: [10.1007/BF00384623](https://doi.org/10.1007/BF00384623) (see page 25).
- [Liu+23] S. Liu et al. **Simultaneous double membrane electrolysis of sodium sulfate solution and preparation of hydroxide precursor for layered cathode materials**. *Journal of Cleaner Production* 428 (Nov. 2023), 139272. ISSN: 09596526. DOI: [10.1016/j.jclepro.2023.139272](https://doi.org/10.1016/j.jclepro.2023.139272) (see page 21).

- [PBS11] R. Pakmor, A. Bauer, and V. Springel. **Magnetohydrodynamics on an unstructured moving grid**. *Monthly Notices of the Royal Astronomical Society* 418:2 (Dec. 1, 2011), 1392–1401. ISSN: 0035-8711. DOI: [10.1111/j.1365-2966.2011.19591.x](https://doi.org/10.1111/j.1365-2966.2011.19591.x) (see page 1).
- [Raf+18] M. Raffel et al. **Particle Image Velocimetry: A Practical Guide**. Springer, Apr. 3, 2018. 680 pp. ISBN: 978-3-319-68852-7 (see pages 26, 29).
- [Sha49] C. Shannon. **Communication in the Presence of Noise**. *Proceedings of the IRE* 37:1 (Jan. 1949), 10–21. ISSN: 2162-6634. DOI: [10.1109/JRPROC.1949.232969](https://doi.org/10.1109/JRPROC.1949.232969) (see page 26).
- [SLS07] U. Shavit, R. J. Lowe, and J. V. Steinbuck. **Intensity Capping: a simple method to improve cross-correlation PIV results**. *Experiments in Fluids* 42 (Feb. 1, 2007), 225–240. ISSN: 1432-1114. DOI: [10.1007/s00348-006-0233-7](https://doi.org/10.1007/s00348-006-0233-7) (see page 24).
- [VZV77] F. R. Vukajlović, S. Zeković, and V. Veljković. **Resistivity calculations for liquid metals**. *Physica B+C* 92:1 (Aug. 1, 1977), 66–72. ISSN: 0378-4363. DOI: [10.1016/0378-4363\(77\)90149-8](https://doi.org/10.1016/0378-4363(77)90149-8) (see page 13).
- [WS05] J. Westerweel and F. Scarano. **Universal outlier detection for PIV data**. *Experiments in Fluids* 39:6 (Dec. 1, 2005), 1096–1100. ISSN: 1432-1114. DOI: [10.1007/s00348-005-0016-6](https://doi.org/10.1007/s00348-005-0016-6) (see page 29).
- [Zem18] J. Zemánek. **Distributed manipulation by controlling force fields through arrays of actuators**. PhD thesis. Faculty of Electrical Engineering, Czech Technical University in Prague, 2018. URL: https://support.dce.felk.cvut.cz/mediawiki/images/9/9f/Diz_2018_zemaneck_jiri.pdf (see pages 6, 7).