NTNU
Norwegian University of
Science and Technology

Measuring gas-liquid mass transfer in very small bubbles

Fredrik Kindsbækken

Chemical Engineering and Biotechnology
Submission date: June 2018
Supervisor: Hugo Atle Jakobsen, IKP

Norwegian University of Science and Technology
Department of Chemical Engineering

Summary

Modern industry is always striving for process optimization and reducing emissions, which in many cases requires an accurate modeling of interfacial gas-liquid mass transfer. To study the interfacial gas-liquid mass transfer in very small gas bubbles, a working rig has been prepared to capture high resolution images of rising gas bubbles with diameters less than 2 mm.

This report gives a basic presentation of theories from literature on the formation of very small bubbles, their departure from the tip of a needle and their rise velocity through the column. Subsequently the equipment used in experiments and how to use them both manually and through the controlling software LabVIEW is explained in detail.

Analysis of the results from bubble production tests and attempts to follow the produced bubbles with a camera, concludes that the current setup is only capable of following bubbles up to a diameter of 0.9 mm. A new camera lift with twice the maximum velocity has been ordered, which should be more than fast enough to follow bubbles with diameters up to 2 mm. The smallest achievable bubble size with the current setup is an equivalent diameter of ∼0.6 mm, assuming the bubbles are perfect spheres. Suggestions on how to reduce the minimum bubble size are given and initiated, but have not been tested due to a lack of time.

Sammendrag

Moderne industri streber alltid mot optimalisering av prosesser og reduksjon av klimagassutslipp, noe som i mange tilfeller krever nøyaktig modellering av masseoverføring i gas-væske reaksjoner. For å studere masseoverføringen fra veldig små gassbobler i væsker, har en fungerende rigg for å ta bilder i høy oppløsning av stigende bobler med diametre opp til 2 mm i en væske blitt laget.

Rapporten gir en enkel presentasjon av teorier fra litteraturen om dannelsen av veldig små bobbler, hvordan de løsriver seg fra tuppen av en nål og hastigheten på boblene gjennom kolonnen. Deretter er utstyret som brukes i forsøkene og hvordan de brukes både manuelt og gjennom bruk av styrings-software LabVIEW beskrevet i detalj.

Analysen av resultatene fra tester med bobleproduksjon og forsøk på å følge boblene med kamera, konkluderer med at det nåværende oppsettet kun er kapabelt til å følge bobler med en diameter opp til 0.9 mm. En ny kameralift med dobbelt så høy max. hastighet er bestilt, noe som skal være mer enn nok til å følge bobler med diametre opp til 2 mm. Den minste oppnåelige boblestørrelsen med det nåværende oppsettet er en ekvivalent diameter på $\sim$0.6 mm, med antagelsen om at boblene er sfæriske. Forslag til hvordan enda mindre bobler kan produseres er gitt i rapporten og er igangsatt, men forslagene har ikke blitt testet på grunn av tidsmangel.

Preface

This Master's thesis was written during the spring of 2018. It concludes the 2 year Master's program in Chemical Engineering at the Norwegian University of Science and Technology, leading to the degree of M.Sc. in Chemical Engineering. The work has been done as a part of the research group in Environmental Engineering and Reactor Technology within the Department of Chemical Engineering. The thesis is a continuation of the specialization project carried out during the fall of 2017.

I would like to thank my supervisor, Professor Hugo A. Jakobsen and co-supervisor Jannike Solsvik for your guidance and handling the ordering of all equipment.

In addition I would like to thank Mikael Hammer and Nicholas La Forgia for assisting me with the software programming and connecting to the hardware.

Also thank you Ida for your support and advice over the last two months. I hope the rig provides the results you need.

I declare that this is an independent work according to the exam regulations of the Norwegian University of Science and Technology (NTNU).

Trondheim, June 20, 2018
Fredrik Kindsbækken

Contents

List of Tables

List of Figures

Nomenclature

AVI	=	Audio Video Interleave
BMP	=	Bitmap
CAT	=	Category
DLL	=	Dynamic Link Library
IPv	=	Internet Protocol version
NI	=	National Instruments
SDK	=	Software Development Kit
VISA	=	Virtual Instrument Software Architecture
VI	=	Virtual Instrument (LabVIEW program)

Abbreviations

a	=	needle radius
d	=	Diameter
$\mathrm{d_e}$	=	Diameter of a spherical bubble of equivalent volume
$\mathrm{D_n}$	=	Needle diameter
g	=	Gravitational acceleration
$\mathrm{D_n}$	=	Needle diameter
L	=	Effective capillary length
$\mathrm{L_{crit}}$	=	Critical effective capillary length
p	=	pressure
$\mathrm{p_{atm}}$	=	Atmospheric pressure
Q	=	Volumetric flow rate
$\mathrm{Q_{crit}}$	=	Critical volumetric flow rate at the tip of needle
$\mathrm{R_f}$	=	Bubble radius where buoyancy and surface tension forces are balanced
udu	=	Custom unit of length (smallest increment of movement for camera lift)
$\mathrm{V_f}$	=	Static bubble volume (corresponding to $\mathrm{R_f}$)
$\mathrm{V_t}$	=	Chamber volume
$\mathrm{v_T}$	=	Terminal velocity
z	=	Water depth at bubble position
σ	=	Surface tension
ρ	=	Density
$\Delta\rho$	=	Difference in density between liquid and gas
μ_l	=	Dynamic viscosity of liquid

Chapter 1

Motivation

Understanding the mass transfer between gas bubbles and a liquid is of great importance in many areas of chemical engineering, with applications in bio-reactors, food processing, water treatment, and multi-phase reactors in general. Utilizing natural gas from oil production, which would otherwise be flaired at the production site, in bio-protein production could be a great way to lower CO_2 emissions and reducing the environmental footprint of food production. As single cell protein production is limited by mass transfer (4), a good model for mass transfer is crucial for the optimization of this process.

In environmental engineering, most processes are involved in interfacial reactions, where according to Chiang and Pan (5) mass transfer is an extremely essential component for system optimization. Historically the design of systems involving mass transfer operations has largely been based on empirical test data from laboratory work or pilot plant scales (6), and "*Understanding of the mass transfer behaviour in bio-reactors for gas treatment will result in improved reactor designs, reactor operation, and modeling tools, which are important to maximize efficiency and minimizing cost.*" Krakman et.al.(7). Gas treatment by biological methods is rapidly gaining acceptance as a more environmental friendly alternative to more conventional techniques, such as chemical scrubbing and adsorption of pollutants. An increasing interest in processes involving interfacial reactions makes studies on mass transfer behaviour highly relevant within the field of chemical engineering. As stated by Kraakman et.al(7): "*Many of the fundamental processes in biological gas treatment systems like mass transfer still require research.*" This statement summarizes the motivation behind this thesis, which is to study mass transfer in gas bubbles rising through a stagnant liquid.

According to Doran (8) "*Bubble size also affects the value of* $\mathrm{k_L}$. *In most fermentation broths, if the bubbles have diameters less than 2 to 3 mm, surface tension effects dominate the behaviour of the bubble surface. As a result, the bubbles behave as rigid spheres with immobile surfaces and no internal gas circulation. A rigid bubble surface gives lower* $\mathrm{k_L}$ *values;* $\mathrm{k_L}$ *decreases with decreasing bubble diameter below about 2 to 3 mm.*" This lack of internal gas circulation would cause the concentration at the gas-liquid interface to equilibrate quickly. As the gas supply to the gas-liquid interface of such a bubble would rely on the relatively slow molecular diffusion rather than mixing, $\mathrm{k_L}$ is expected to be lower.

The purpose of this thesis is to make a working rig, which allows production of bubbles in the rigid sphere domain and collecting data by capturing high-speed, high-resolution images of the bubbles while rising through a liquid. The data collected from experiments on this rig will be used in further studies to develop models for determining the mass-transfer coefficient for bubbles in the rigid sphere domain, and thereby confirming or disproving Doran's assumptions.

If the mass transfer coefficient turns out not to decrease significantly for very small bubbles, mass transfer would increase due to a higher surface to volume ratio, which is important for system optimization.

Chapter 2

Basic Theory

Producing and recording images of very small bubbles (<2 mm) rising through a stagnant liquid requires knowledge on the dynamics of bubble growth and departure from the needle. Several relations crucial to a reliable production of very small bubbles has previously been studied in the works of Oguz and Prosperetti (1) and Vejrazka et.al (9), which have led to a number of equations (2.1, 2.3, 2.5 Ozug and 2.4, 2.6, 2.7 Vejrazka) presented in this chapter. These relations are used to determine conditions like needle diameter, gas-flow rate, chamber volume and capillary length required to achieve a reliable production of bubbles with the desired size.

2.1 Bubble growth and detachment from a needle

For a slowly forming bubble to detach from the needle, the surface tension forces holding the bubble in place needs to be exceeded by buoyancy forces generating lift. The bubble radius where buoyancy and surface tension forces are in balance is determined by:

$$\mathrm{R_f} = (\frac{3\sigma \mathrm{a}}{2\rho_\mathrm{l}\mathrm{g}})^{\frac{1}{3}} \tag{2.1}$$

Where σ is the surface tension between the gas and liquid, a is the needle radius, ρ_l is the liquid density and g is gravitational acceleration. The static bubble volume, which is infinitesimally lower than minimum achievable volume of a spherical bubble then becomes:

$$\mathrm{V_f} = \frac{4\pi}{3}\mathrm{R_f^3} \tag{2.2}$$

To achieve a quasi-static bubble formation where the bubble volume V is approximately equal to $\mathrm{V_f}$ requires a flow rate lower than the critical flow rate $\mathrm{Q_{crit}}$ to avoid the effects of inertial forces.

$$\mathrm{Q_{crit}} = \pi(\frac{16}{3\mathrm{g}^2})^{\frac{1}{6}}(\frac{\sigma \mathrm{a}}{\rho_\mathrm{l}})^{\frac{5}{6}} \tag{2.3}$$

In Figure 2.1 the transition between the quasi-static bubble formation dominated by surface tension forces, and the inertia domain becomes clear as the flow rate exceeds $\mathrm{Q_{crit}}$

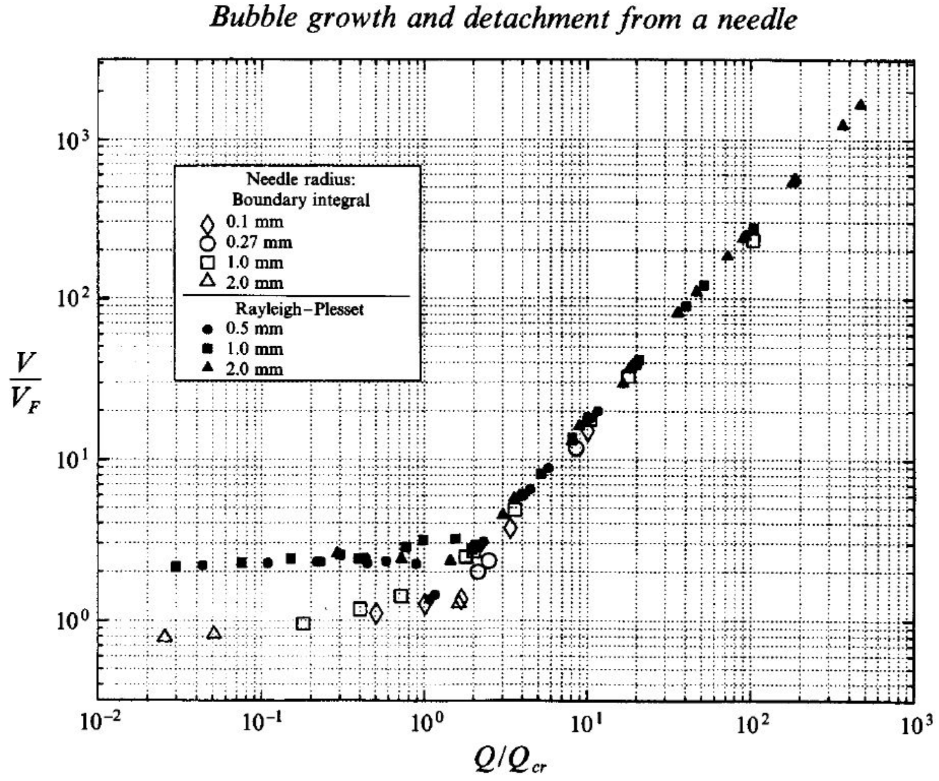


Figure 2.1: Plot of bubble size vs. injection flow rate. Reprinted from Ozug (1)

For the instantaneous flow rate to correspond to the syringe pump inlet flow rate, the chamber volume (tube, syringe and wide section of needle) $\mathrm{V_t}$ needs to be lower than a critical volume calculated by eq.2.4

$$\mathrm{V_{crit}} = \frac{\pi \mathrm{p}_\infty \mathrm{D_n^2}}{4\mathrm{g}\rho_\mathrm{l}} \tag{2.4}$$

Chamber volumes larger than $\mathrm{V_{crit}}$ allows the gas to compress to a degree which causes the instantaneous flow rate through the needle to exceed pump inlet flow when producing bubbles. This can cause problems when producing single bubbles. For a bubble to form, the chamber pressure needs to exceed the minimum value:

$$\mathrm{P_{C,min}} = \mathrm{P}_\infty + \frac{2\sigma}{\mathrm{a}} \tag{2.5}$$

For small needles this over-pressure is large. Combined with a high degree of gas compression due to $\mathrm{V_t} >> \mathrm{V_{crit}}$, several bubbles are rapidly produced as the chamber pressure decreases.

In the case of $\mathrm{V_t} >> \mathrm{V_{crit}}$, the condition for these sudden bursts of rapid bubble injections can be written as:

$$\frac{\mathrm{Q}}{\mathrm{Q_{crit}}} \cdot \frac{\mathrm{L}}{\mathrm{L_{crit}}} >> 1 \tag{2.6}$$

Where the critical capillary length $\mathrm{L_{crit}}$ is:

$$\mathrm{L_{crit}} = \frac{6^{1/6}}{32} \cdot (\frac{\sigma \mathrm{D_n^{13} g^2} \rho^5}{\mu_\mathrm{g}^6})^{1/6} \tag{2.7}$$

2.2 Bubble velocity

Most of the bubble size range of interest (<2 mm) lies within the spherical regime. As shown in figure 2.2 the terminal velocity varies heavily with bubble size in this range, with velocities ranging from 2 cm/s up to ~ 35 cm/s in pure water. Knowing the exact bubble size will therefore be crucial to the ability to follow rising bubbles with a camera.

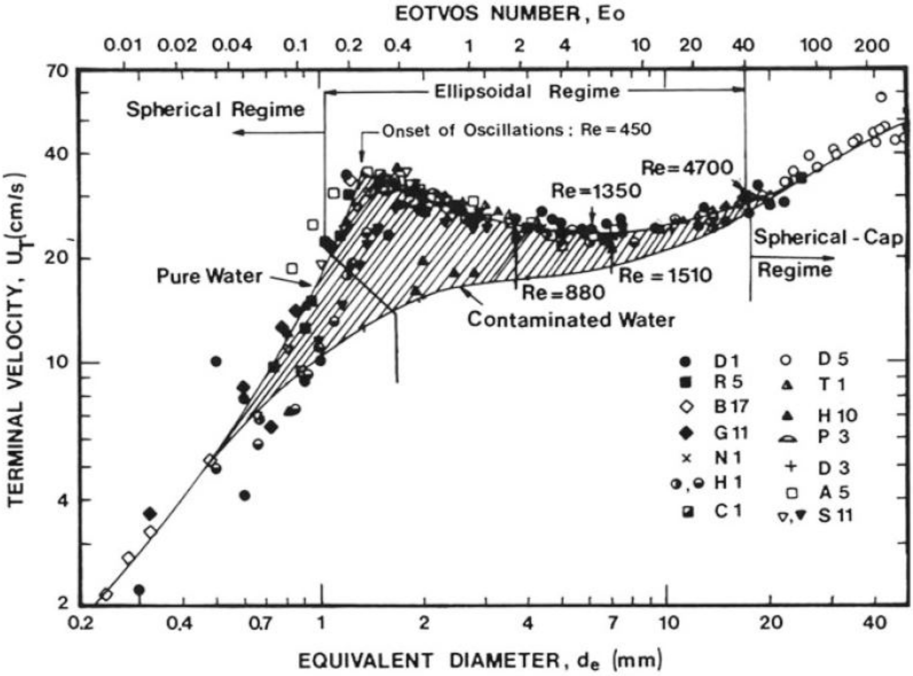


Figure 2.2: Terminal velocity of air bubbles in water at 20°C. Reprinted from Hugo A. Jakobsen (2)

A good model for calculating the terminal velocity of a bubble rising in a pure liquid presented by Baz-Rodrigues et al. (10) is presented in equations 1 - 4

$$\mathrm{V_T} = \frac{1}{(\frac{1}{\mathrm{V_{T1}}}\frac{1}{\mathrm{V_{T2}}})^{1/2}} \tag{2.8}$$

where the rise velocity dominated by viscous effects (small bubbles) is:

$$\mathrm{V_{T1}} = \mathrm{V_{T,pot}}[1 + 0.73667\frac{(\mathrm{gd_e})^{1/2}}{\mathrm{V_{T,pot}}}]^{1/2} \tag{2.9}$$

and the terminal velocity for a bubble in potential flow is:

$$\mathrm{V_{T,pot}} = \frac{\Delta\rho \mathrm{gd_e^2}}{36\mu_\mathrm{l}} \tag{2.10}$$

and the corresponding rise velocity dominated by surface tension effects (larger bubbles) is:

$$\mathrm{V_{T2}} = (\frac{3\sigma}{\rho_\mathrm{l}\mathrm{d_e}} + \frac{\mathrm{gd_e}\Delta\rho}{2\rho_\mathrm{l}})^{1/2} \tag{2.11}$$

$\mathrm{d_e}$ is the diameter of a spherical bubble of equivalent volume ($\mathrm{d_e} = \mathrm{d}$ in the spherical domain), $\Delta\rho$ is difference in density between the liquid and gas. μ_l is the liquid dynamic viscosity, g is gravitational acceleration and σ is the surface tension. The pressure drop over the column causes an increase in volume, which affects the terminal velocity of the bubble. At a water depth of 2 m, which is used in the experimental setup, the pressure drop causes bubble volume to expand by 18,9 % from the bottom to the top of the column assuming no mass transfer at constant temperature ($\mathrm{p_1V_1} = \mathrm{p_2V_2}$). Pressure at a given depth is given by equation 2.12.

$$\mathrm{p_{(z)}} = \mathrm{p_{atm}} + \rho_\mathrm{l}\mathrm{gz} \tag{2.12}$$

Where $\mathrm{p_{atm}}$ is atmospheric pressure, ρ_l is liquid density, g is gravitational acceleration and z is water depth at bubble position. For the largest bubbles in the region of interest (d=2 mm) the diameter increases from 2,00 mm to 2,12 mm, which decreases the velocity by 0.51 cm/s according to equation 1 . For the smaller bubbles (eg. d=0,5 mm) the diameter increases from 0,50 mm to 0,53 mm, which increases the velocity by 0.97 cm/s according to equation 1. The resulting changes in bubble velocity could be a problem for the recording of the smallest bubbles when there is no mass transfer, with a camera moving at constant velocity.

To produce bubbles in the spherical regime the gas flow rate needs to be below a critical value $\mathrm{Q_{crit}}$ and the needle size needs to be sufficiently small. As long as the gas flow rate is below the critical value, bubble sizes are independent of flow rates as shown in Figure 2.1.

2.3 Mass transfer models

The mass transfer coefficient express how fast a component is moving through the gas-liquid interface, which is a key parameter in the design of bioreactors. Several different correlations has been developed to estimate the mass transfer coefficient for interfacial mass transfer processes. All these correlations are developed with a basis in one of the four following theories on how the mass transfer dynamics works:

2.3.1 Film Theory

Film theory is a very simplified mass transfer model, which assumes there is a stagnant film on both sides of the gas-liquid interface as illustrated in figure 2.3. The bulk is perfectly mixed and concentration differences occur over the gas and liquid film. The gas-liquid interface is assumed to be at an equilibrium, where the mass transfer flux is given by a steady molecular diffusion.

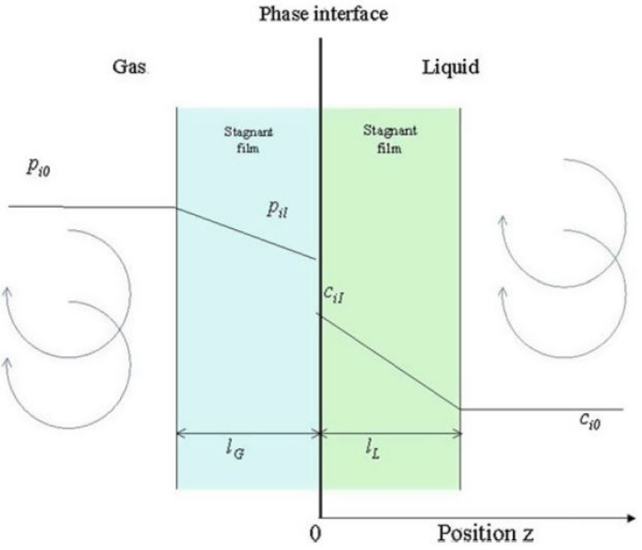


Figure 2.3: Illustration of gas-liquid interface as presented by film theory. Figure reprinted from Hugo A. Jakobsen (2)

2.3.2 Penetration Theory

In penetration theory the mass transfer is assumed to be an unsteady state mass transfer to liquid elements in contact with the bubble surface. The liquid elements attaches to the top of the bubble and slides down along the gas-liquid interface as the bubble rises. At the bottom of the bubble they detach and mixes with the bulk liquid. The elements contact time with the bubble surface is assumed to be the same for all elements. Figure 2.4 illustrates how the liquid elements moves along the bubble surface.

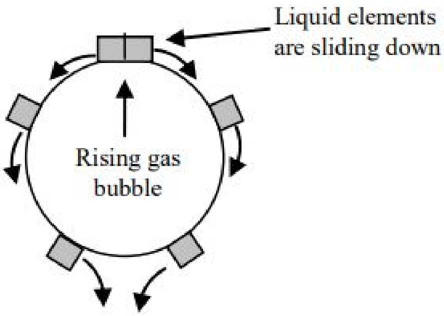


Figure 2.4: The attachment, movement and detachment of liquid elements at the bubble surface. Figure reprinted from NPTEL lecture on mass transfer (3)

2.3.3 Surface Renewal Theory

Surface renewal theory is an improvement of the penetration theory, which assumes that a portion of the liquid elements at the gas-liquid interface are replaces by fresh elements due to the motion of eddies near the bubble surface. The replacement of liquid elements caused by turbulent motion causes the assumption that all elements have the same residence time at the bubble surface to no longer be valid. Figure 2.5 illustrates the replacement of a liquid element residing at the gas-liquid interface.

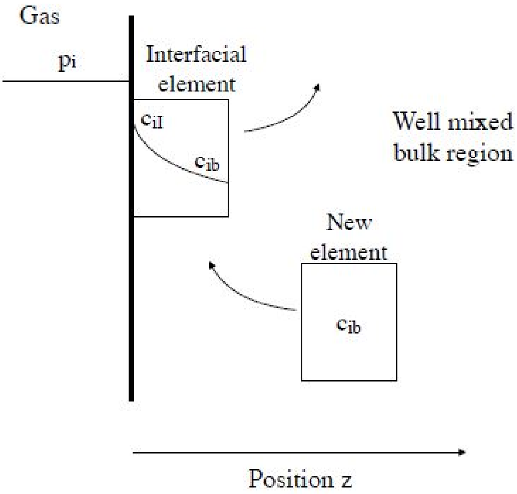


Figure 2.5: Illustration of the replacement of a liquid element residing at the gas-liquid interface. Figure reprinted from Ida K. Kure (4)

2.3.4 Boundary Layer Theory

Boundary layer theory is more complex and the most comprehensive of the four mass transfer models. Boundary layer theory takes into account the hydrodynamics that characterizes the system to give a more realistic representation of mass transfer dynamics. The heat and mass transfer equation is directly coupled with the momentum balance, which results in analytical solutions the are more accurate compared to the film or surface renewal theories. However, only idealized flow situations can be considered in order to solve the governing equations.

Chapter 3

Equipment and hardware setup

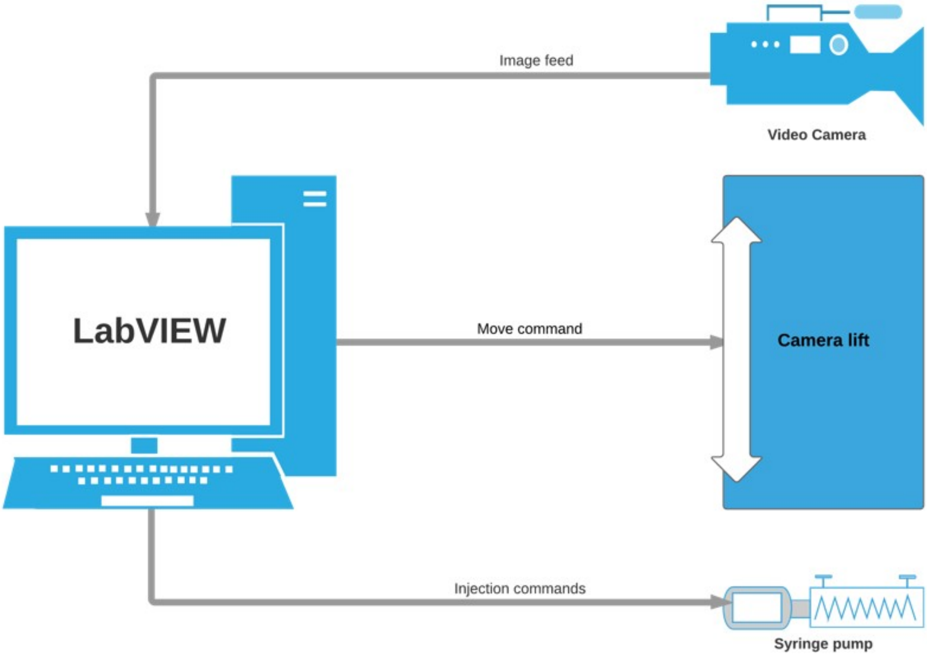


Figure 3.1: Hardware flowchart

This chapter is an introduction to the equipment used to acquire high resolution images of single bubbles rising through a liquid. Instructions on how to set up equipment, operate apparatuses manually and do the necessary preparation for software control are described in this chapter. Further instructions on how to run experiments through LabVIEW is described later in chapter 4

3.1 Column

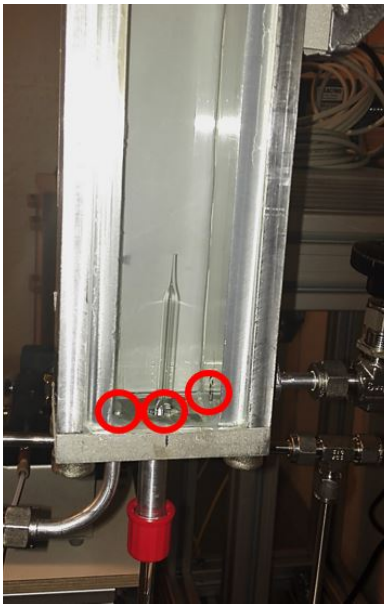


Figure 3.2: Lower part of column. The bottom inlets are circled in red.

A 4x4x220 cm column with two glass walls on opposing sides is where gas bubbles are injected into the liquid. There are three inlets/outlets at the bottom of the column and two at the top. The first bottom inlet is a hole at the center, where the needle is placed to inject gas bubbles. The second inlet/outlet is where liquid is pumped in and drained from out the column (to the left in figure 3.2). The valves determining whether a liquid enters or exits the column are located on the side opposing the camera. Make sure both valves are in the closed position when running experiments, and operate the valves as illustrated in figure 3.3. while pumping in/ draining out liquid. The third inlet is at the back of the column facing the wall (to the right in figure 3.2). This inlet is for injecting inert gasses to make sure the liquid is saturated with gas before running experiments. The outlet at the top of the column is for ventilation to prevent pressure from building up when degassing the liquid. At the back of the column facing the wall, near the top there is an input for a gas measuring and detection device. If a measurable amount of gas has accumulated in the liquid, the water needs to be changed. The outlets/ inlets at the top of the column should be sealed when not in use, to prevent air and dust particles from entering the column. Make sure to drain the column and let it dry off every day to prevent unwanted growth on the inside of the column walls.

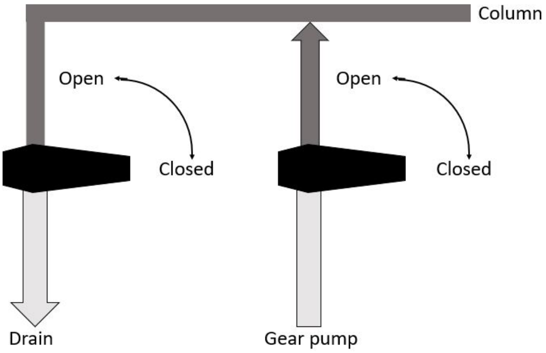


Figure 3.3: Valve operations for draining and filling the column

3.2 Gear pump

To pump liquids in to the column, a Reglo-Z Digital gear pump manufactured by ISMATEC® is used. This pump is operated manually by using the interface on the pump shown in Figure 3.4.

Table 3.1: Operating panel functions

Number	Function
1.	On/Off switch
2.	Selecting operating mode
3.	LED display for operating modes
4.	Digital display
5.	Increase value
6.	Reduce value
7.	Start and stop the pump
8.	Change pump direction (PUMP mode) / Reset TOTAL volume (TOTAL mode)
9.	Calibrate flow rate or volume (click)/ Set speed to max. (hold while running)

Numbering (#) refers to component numbers from figure 3.4 and corresponding table 3.1. While the pump is turned on (1), press the MODE button (2) until the LED display (3) for PUMP mode (flow rate) lights up. Then select the desired flow rate in [mL/min] using the increase/decrease value buttons (5) and (6). Recommended flow rate is no more than 600 mL/min, as potential air bubbles have a tendency to disperse and stick to the glass

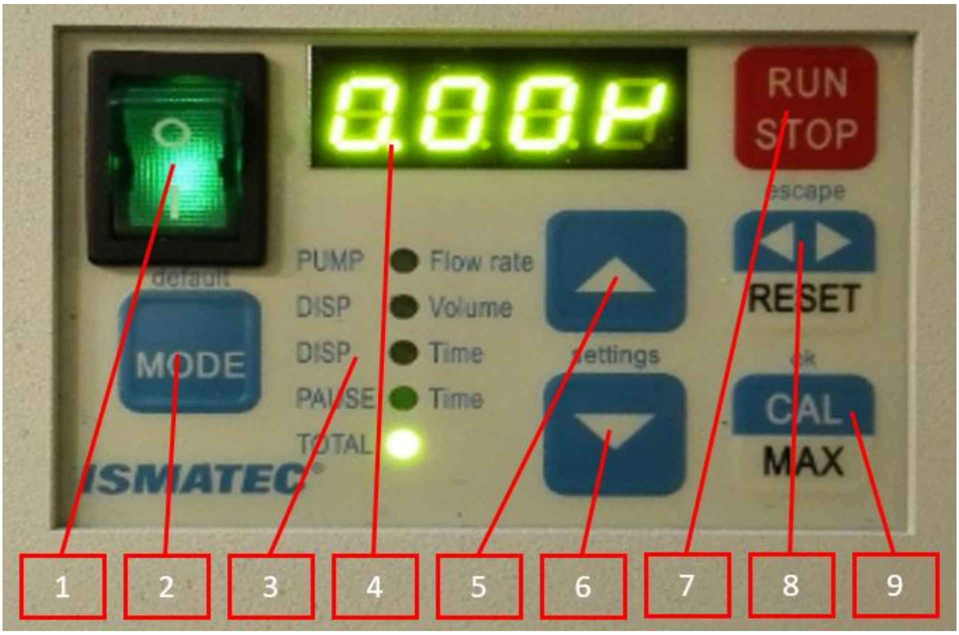


Figure 3.4: Gear pump operating panel

at higher flow rates (more turbulent flow). When running the pump leave both valves for filling and draining the column open, until all air bubbles in the tubes are gone. This way air bubbles are prevented from entering the column, dispersing and then sticking to the glass. Stop the pump (7) and press reset before closing the drain valve in order for the accumulated volume on the digital display (4) to neglect the amount pumped out into the drain. To see how much liquid has been pumped in, press the MODE button (2) until the LED display for TOTAL lights up as shown in 3.4. In TOTAL mode the digital display shows the accumulated volume pumped in to the column. Remember to close the valve after filling the column to prevent the liquid from running back down through the pump.

3.3 Syringe pump

A Legato200® step motor syringe pump, manufactured by KdScientific® is used to inject gas bubbles in to the column. The pump is controlled in LabVIEW through an USB connection to the computer, or operated manually through the touch screen interface.

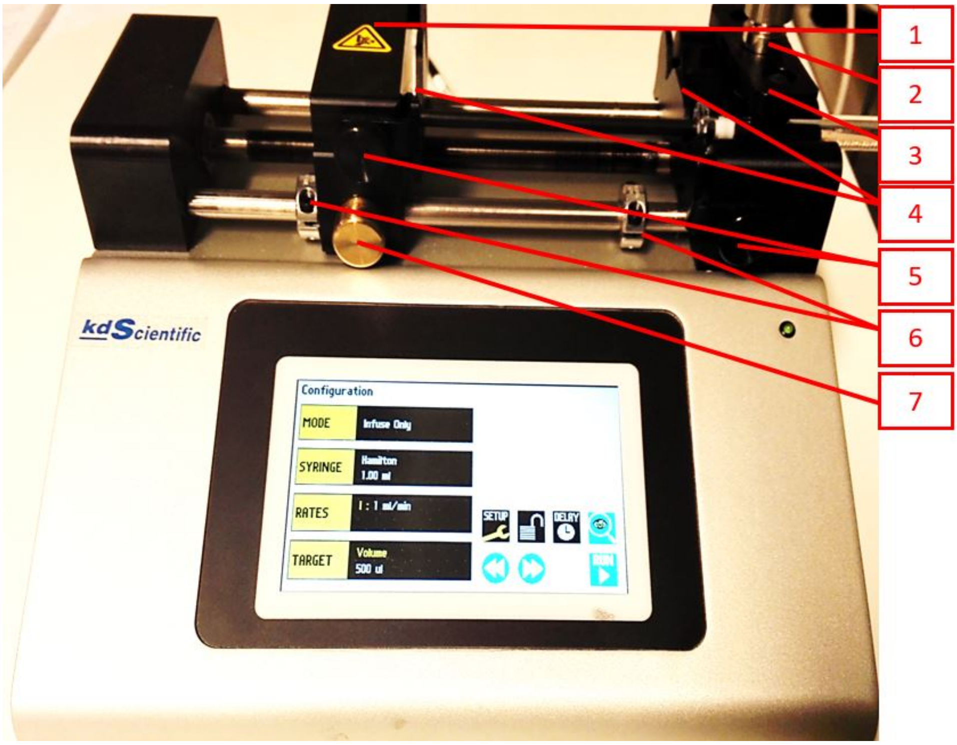


Figure 3.5: Legato 200 syringe pump

Table 3.2: Syringe pump components

Number	Function
1.	Pusher Block
2.	Syringe Barrel Clamp
3.	Syringe Barrel Holder
4.	Retaining Brackets
5.	Bracket Clamp Knobs
6.	Mechanical Stop Collars
7.	Release Knob

Numbering (#) refers to component numbers from figure 3.5 and corresponding table 3.2. When mounting a syringe on the pump, make sure both the barrel flange and syringe plunger are placed in the retaining brackets (4) before fastening the syringe barrel clamp (2). Be gentle when fastening the syringe barrel clamp, as too much pressure may break the syringe. If the syringe does not fit in the retaining brackets, loosen the bracket clamping knobs (5) and adjust the retaining brackets position. Before operating the syringe pump, make sure the rear mechanical stop collar (6) prevents the plunger from being pulled all the way out of the syringe. Also make sure the mechanical stop collar (6) in front of the pusher block, prevents the plunger from reaching the front of the syringe, which might break the syringe. Press and hold the release knob (7) to move the pusher block(1) freely. To re-position the mechanical stop collars, loosen the screws with an Allen key.

When operating the syringe pump manually in the touch screen interface, open ”Setup” and make sure the pusher block force ”Force” is set to 50% or lower as a contingency in case the mechanical stop collars aren’t properly fastened. At 50% the maximum linear force applied before stopping the pump is 17 kg force, which should not be enough force to harm the syringe, unless already damaged. After setting the force level, press ”ENTR” to get back to the configuration screen. To choose pump direction, press ”MODE” and select ”Infuse only” or ”Withdraw only” depending on the purpose of the operation, then press ”ENTR” to set the direction and get back to the configuration menu. To configure syringe dimensions press ”SYRINGE” in the configuration menu, and select the syringe mounted on the pump eg.”Hammilton 100” then choose the syringe volume eg.”1mL”. If the mounted syringe is not listed in the pump’s library, choose ”Custom” and add the syringe diameter and volume manually. Once the correct syringe dimensions are configured, set the flow rate by pressing ”RATES” in the configuration menu and assigning injection/withdrawal rate (Pusher travel rate min - max is 0.36 µm/min - 190.80 mm/min). To change flow rate units press the block with the currently active units, and select appropriate units for volume and time from the menu. The amount of gas to inject is chosen by pressing ”TARGET” in the configuration menu, which opens the ”Set Target Volume or Time” menu. Choose ”Volume”, select appropriate units and enter the target volume. When configuration is completed you can preview the operation by pressing the magnifying glass on the right side of the screen. Press the ”RUN” button in the bottom right corner to start injecting/ withdrawing.

To communicate with the pump in LabVIEW, you need to know which communications port to call. To find the communications port, open ”Device Manager” from the Windows search function, and select ”Ports (COM & LPT)”. The communications port is called ”USB Serial Device (COM#)”. The address inside the parenthesis is the ”VISA resource name” you need to call in LabVIEW (eg. COM4). If unsure on which port is connected to the syringe pump try disconnecting the USB connection, and observe which of the communications ports disappears. The one that disappears is the port you need to call in the LabVIEW program. When refilling the syringe with gas, the column needs to be drained of liquid to prevent water from intruding in to the needle. A solution to this problem is the use of a one-way-valve or a T-valve, but it has not been successfully implemented yet.

3.4 Camera lift

The lift controller is connected to the computer trough the RS232 port shown in figure 3.10, and controlled using LabVIEW. The manufacturer ISEL® also provides their own software (PAL-PC), which can be used to run predetermined sequences with the lift, but is not compatible with other equipment used in this setup. Controller parameters eg. maximum current, velocity, acceleration and PID regulator parameters are set manually in ”ACsetup269”, which is found in the attached software file. The lift can be manually operated through the controller’s interface (Numbering (#) in this section refers to button numbers from to press from figure 3.6. eg. ”Manual (2)” means that to select ”Maunal” you have to press the button which is numbered ”2” in figure 3.6). To manually operate the lift select ”Manual (2)” from the main menu, and select ”Move (2)”. This opens the manual movement menu. In the manual movement menu, press (1) to select operation. The selected operation is indicated by a ”≫” sign pointing to the selected operation in the digital display. To adjust lift velocity, select ”Speed” in the manual movement menu and increase (2) or decrease (3) the speed (displayed velocity units are [mm/s]). When desired lift velocity is set, select ”Move” from the same menu and run the lift by holding in (3) or (2) to move the lift position up or down respectively. Note that the negative direction ”**-**” is up, and the positive direction ”**+**” is down. The lift has a servo motor with a precision of 1 µm at a gear ratio of 1:1. At a gear ratio of 2:1 one move unit equates to 2 µm, and for a gear ratio of 3:1 one move unit equates to 3 µm etc. The maximum velocity of the lift motor is 3000 rpm, with a ball-screw-pitch of 5 mm for 1:1 gears this yield 25 cm/s. Changing to 2:1 gears will yield a maximum velocity of 50 cm/s.

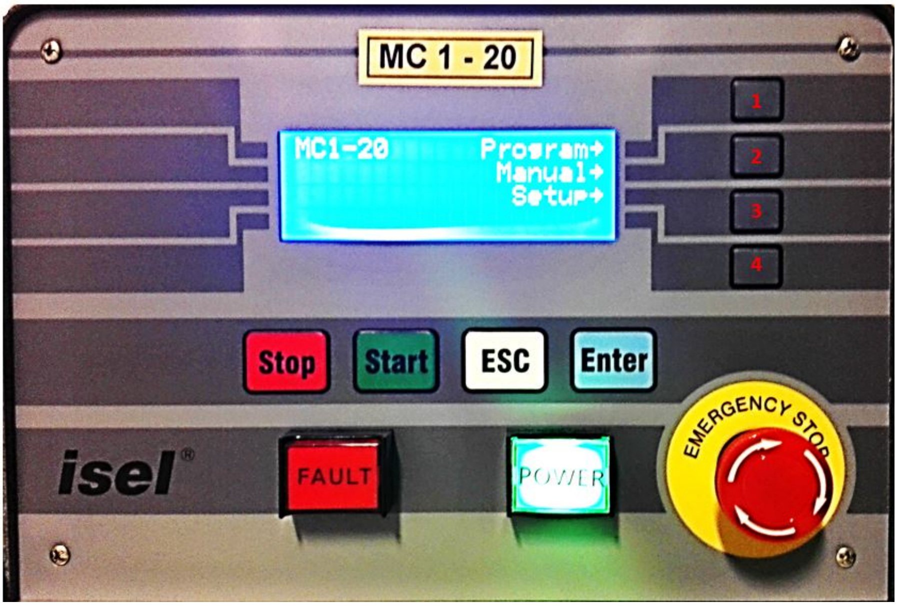
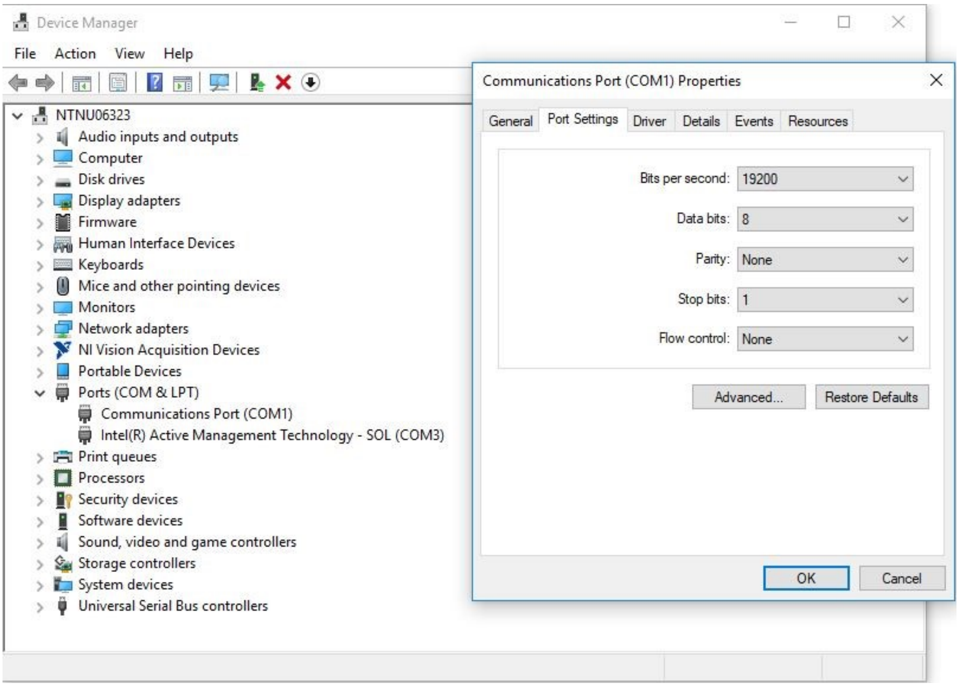


Figure 3.6: Frontside view of lift controller MC 1 - 20 showing the Controller’s interface

ACsetup is a software used to configure controller drivers. Settings used for acquiring bubble images are saved in the controller memory and can be uploaded back to the PC as shown in Figure 3.8.

Before communication between the lift controller and computer can be established, the communications port properties needs to correspond to the controller as shown in Figure 3.7. The property window is opened by right clicking the communications port in device manager as explained in the end of section 3.3 and selecting properties. In the Communications Port (COM #) Properties tab, select ”Port Settings” from the menu at the top to open the window shown to the right in figure 3.7.



Device Manager

File Action View Help

NTNU06323
- Audio inputs and outputs
- Computer
- Disk drives
- Display adapters
- Firmware
- Human Interface Devices
- Keyboards
- Mice and other pointing devices
- Monitors
- Network adapters
- NI Vision Acquisition Devices
- Portable Devices
- Ports (COM & LPT)
  - Communications Port (COM1)
  - Intel(R) Active Management Technology - SOL (COM3)
- Print queues
- Processors
- Security devices
- Software devices
- Sound, video and game controllers
- Storage controllers
- System devices
- Universal Serial Bus controllers

Communications Port (COM1) Properties

General | Port Settings | Driver | Details | Events | Resources

Bits per second: 19200

Data bits: 8

Parity: None

Stop bits: 1

Flow control: None

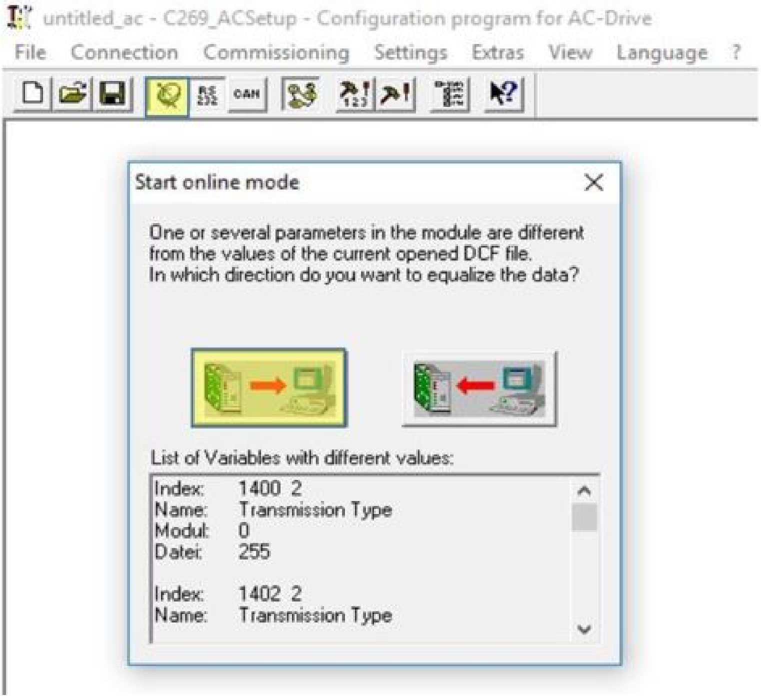
Advanced... | Restore Defaults

OK | Cancel

Figure 3.7: Communications Port settings

With port settings mirroring figure 3.7 and the controller in driver setup mode, the controller parameters can be configured in ”ACsetup269”. ”ACsetup269” can be opened from a shortcut on the desktop, and can otherwise be installed from a backup of the ISEL CD on a flash drive. Driver setup mode is manually activated using the controller display interface. To activate driver setup mode from the start up menu choose ”setup (3)” , then select ”driver(1)” and press the ”start” button. If anything other than the main menu from figure 3.6 is shown on the digital display after turning the power on, press ”ESC” until the start up menu appears.

When connecting to the controller in ”ACsetup269” by clicking the satellite dish symbol ”Go online” highlighted in in figure 3.8, the window ”start online mode” might appear. Click the highlighted image (to the left) to upload the current settings from the controller to the computer. If the connection fails, make sure LabVIEW is not running as the communication might be occupied. If this does not work you should restart the computer as the communications port might be occupied by some other processes running in the background.



One or several parameters in the module are different from the values of the current opened DCF file.
In which direction do you want to equalize the data?

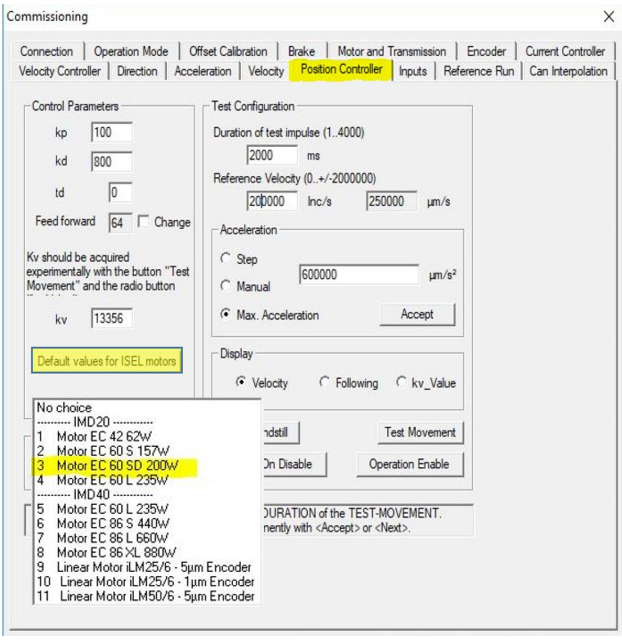
List of Variables with different values:

Index: 1400 2
Name: Transmission Type
Modul: 0
Datei: 255

Index: 1402 2
Name: Transmission Type

Figure 3.8: ACsetup ”Start online mode” window

If the driver settings are reset to default, some parameters needs to be changed in order to successfully run experiments. To change the PID regulator parameters open the ”commissioning” tab from the menu at the top of the screen and select ”Random order”. After opening random ordered commissioning, select the ”Position Controller” tab and click the ”Default values for ISEL motors” button and choose setting number 3 ”Motor EC60 SD 200W” as highlighted in figure 3.9. When the regulator parameters are selected, click ”Apply” in the bottom right corner. To test the lift response, enter the velocity at which the lift should run as ”Reference Velocity”, enter the acceleration and ”Duration of test impulse”. The test movement runs the lift in upwards for 1/4 of the duration, then down for 2/4 of the duration, then up again for 1/4 of the test duration, back to the starting position. Before pressing ”Test movement” to run a test, make sure the lift will not encounter an end switch during the test run and press ”Operation Enable”. There is an option to ”Invert axis” in the ”Direction” tab in commissioning.


Commissioning
Connection | Operation Mode | Offset Calibration | Brake | Motor and Transmission | Encoder | Current Controller
Velocity Controller | Direction | Acceleration | Velocity | Position Controller | Inputs | Reference Run | Can Interpolation
Default values for ISEL motors
Accept
ndstill
Test Movement
On Disable
Operation Enable

Control Parameters
kp 100
kd 800
td 0
Feed forward 64 □ Change
Kv should be acquired experimentally with the button "Test Movement" and the radio button
kv 13356

Test Configuration
Duration of test impulse (1..4000) 2000 ms
Reference Velocity (0..+/-2000000) 200000 Inc/s 250000 µm/s
Acceleration
○ Step
○ Manual 600000 µm/s²
● Max. Acceleration

Display
● Velocity ○ Following ○ kv_Value

No choice
--------- IMD20 ------------
1 Motor EC 42 62W
2 Motor EC 60 S 157W
3 Motor EC 60 SD 200W
4 Motor EC 60 L 235W
--------- IMD40 ------------
5 Motor EC 60 L 235W
6 Motor EC 86 S 440W
7 Motor EC 86 L 660W
8 Motor EC 86 XL 880W
9 Linear Motor iLM25/6 - 5µm Encoder
10 Linear Motor iLM25/6 - 1µm Encoder
11 Linear Motor iLM50/6 - 5µm Encoder

DURATION of the TEST-MOVEMENT.
nently with <Accept> or <Next>.

Figure 3.9: Regulator parameters

The next five parameters are found in ”Settings” from the menu at the top of the screen. Open the ”Object Directory” tab from the ”Settings” menu and expand the list under ”Device Profile Objects”. Select the parameters shown in table 3.3, set their values accordingly and check off ”Update permanently” below the entered value. When all parameters are set, click ”Save all parameters in module” and press the ”OK” button.

Table 3.3: Driver setup parameter values

Directory Number	Name	Value
6073	Max Current (mA)	25’000
6075	Motor Rated Current (mA)	12’000
607F	Max Profile Velocity (udu/s)	250’000
6083	Profile Acceleration (udu/s^2)	3’000’000
60C5	Max Acceleration (udu/s^2)	4’000’000

After completing the driver setup and disconnecting from the controller in ”ACsetup”, press ”Stop” on the controller interface, then press ”ESC” twice to get back to the start up menu. To activate DNC-Mode which allows the lift to be controlled via LabVIEW, select ”Program (1)” then ”DNC-MODE (1)” and ”Start”.

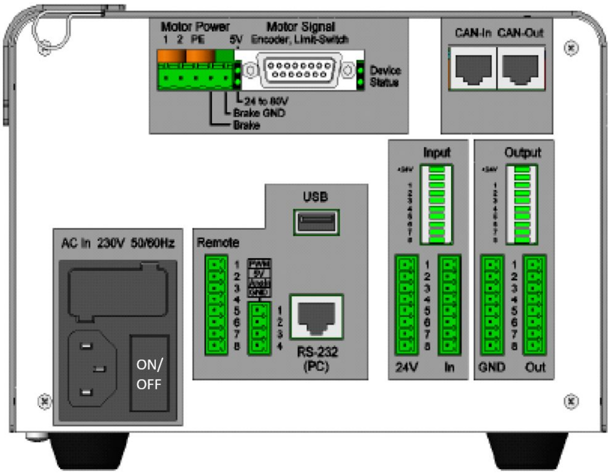


Figure 3.10: Backside view of lift controller MC 1 - 20. Reprinted from MC1-Series Operating Instructions

3.5 Photron fast cam

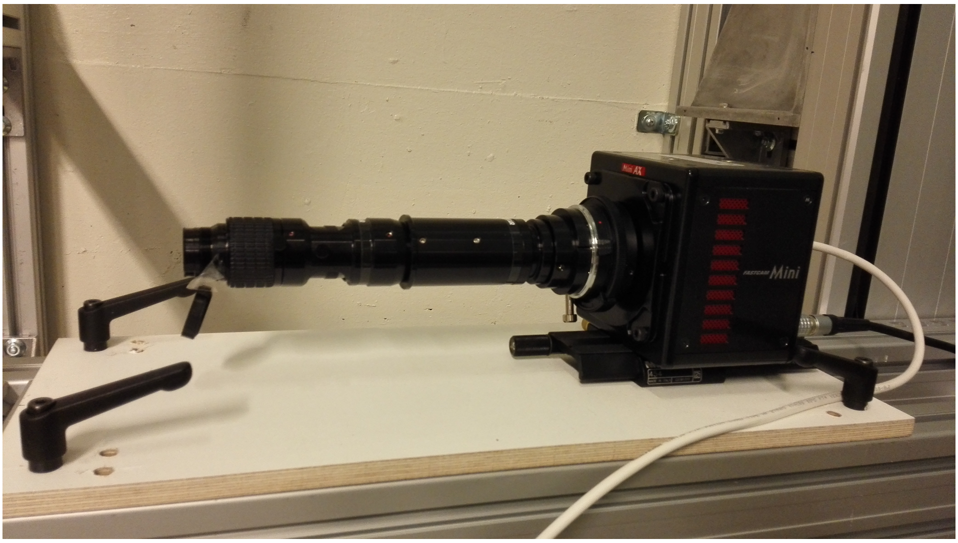
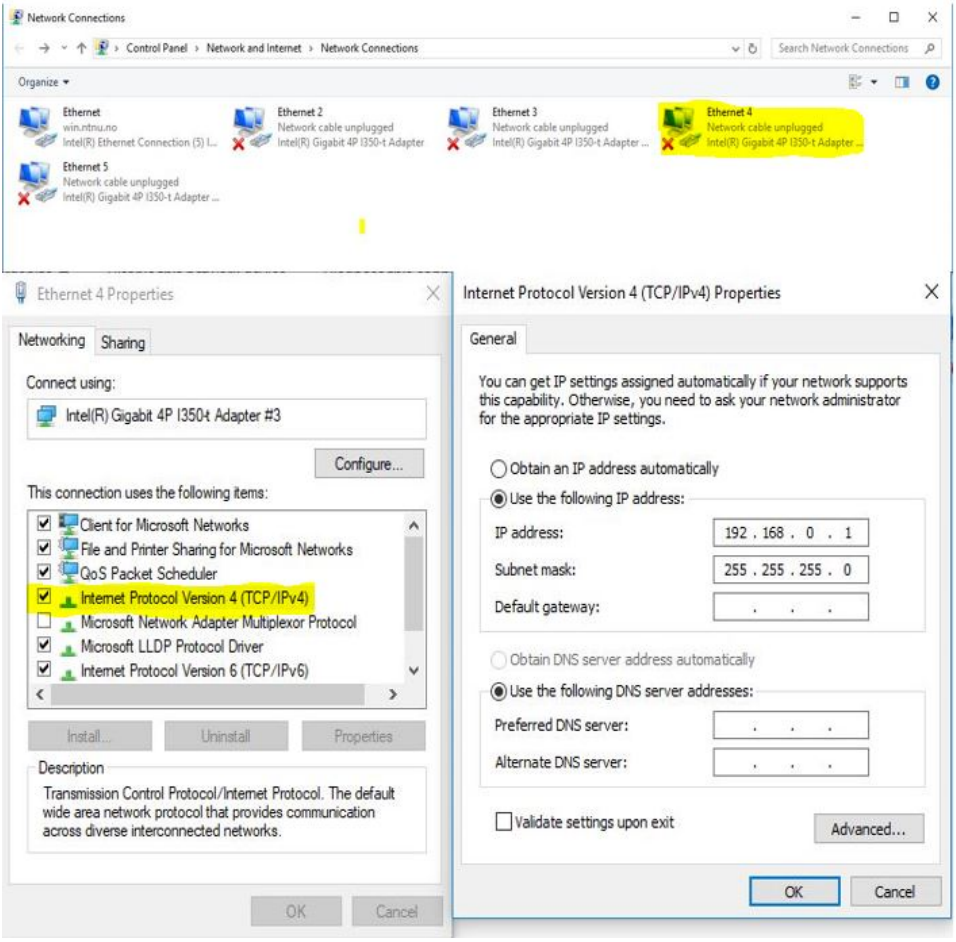


Figure 3.11: Photron FASTCAM mini AX 100

Photron FASTCAM mini AX 100 is used to acquire images of the rising bubbles. The camera captures 1024x1024 pixel images at a rate up to 4000 frames per second. Communication with the computer is done through a CAT6 ethernet cable, which is connected to an ethernet port on the computer. The camera connection is configured on the computer. Configuration of the connection is done in the network connections tab which is found by opening the ”Network and sharing centre” from the computer control panel and selecting ”Change adapter settings” from the menu on the left. To establish communication between the camera and computer, right click the ethernet port connected to the camera and select ”properties”. This opens the properties tab as shown in the bottom left side of figure 3.12 . To find out which of the ethernet ports to click, turn on the camera and observe which port changes status from ”Network cable unplugged” to ”unidentified network”. In the ethernet properties tab, open the properties for ”Internet Protocol Version 4 (TCP/IPv4)” by right clicking it and selecting ”properties”. Select ”Use the following IP address” and enter the IP address shown in figure 3.12. The subnet mask should appear on its own when you enter the IP address. If the subnet mask does not appear after entering the IP address, enter the following subnet mask: 255.255.255.0



Network Connections

Control Panel > Network and Internet > Network Connections

Search Network Connections

Organize ▾

Ethernet
win.ntnu.no
Intel(R) Ethernet Connection (5) I...

Ethernet 2
Network cable unplugged
Intel(R) Gigabit 4P I350-t Adapter

Ethernet 3
Network cable unplugged
Intel(R) Gigabit 4P I350-t Adapter ...

Ethernet 4
Network cable unplugged
Intel(R) Gigabit 4P I350-t Adapter ...

Ethernet 5
Network cable unplugged
Intel(R) Gigabit 4P I350-t Adapter ...

Ethernet 4 Properties

Networking | Sharing

Connect using:
Intel(R) Gigabit 4P I350-t Adapter #3

Configure...

This connection uses the following items:

- ☑ Client for Microsoft Networks
- ☑ File and Printer Sharing for Microsoft Networks
- ☑ QoS Packet Scheduler
- ☑ Internet Protocol Version 4 (TCP/IPv4)
- □ Microsoft Network Adapter Multiplexor Protocol
- ☑ Microsoft LLDP Protocol Driver
- ☑ Internet Protocol Version 6 (TCP/IPv6)

Install... | Uninstall | Properties

Description
Transmission Control Protocol/Internet Protocol. The default wide area network protocol that provides communication across diverse interconnected networks.

OK | Cancel

Internet Protocol Version 4 (TCP/IPv4) Properties

General

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

○ Obtain an IP address automatically
● Use the following IP address:

IP address: 192 . 168 . 0 . 1
Subnet mask: 255 . 255 . 255 . 0
Default gateway: . . .

○ Obtain DNS server address automatically
● Use the following DNS server addresses:

Preferred DNS server: . . .
Alternate DNS server: . . .

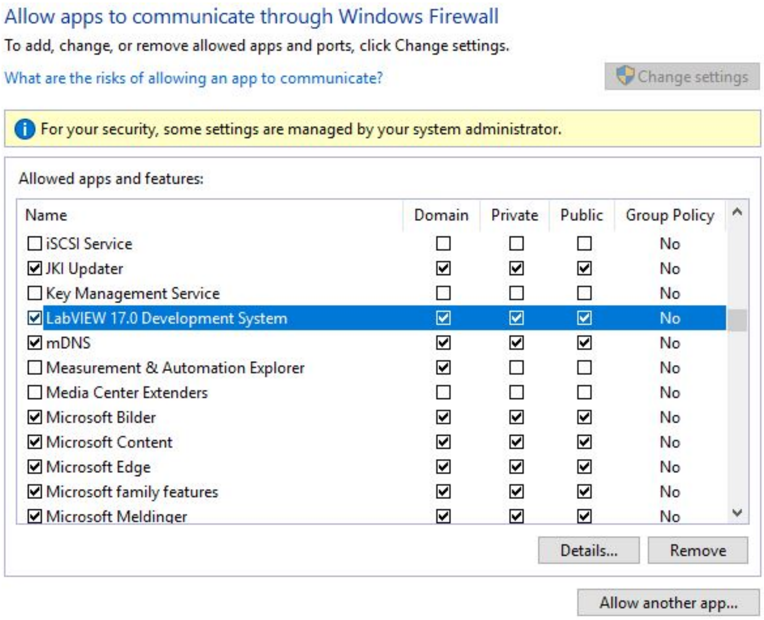
□ Validate settings upon exit

Advanced...

OK | Cancel

Figure 3.12: Camera configuration setup

The camera signal is blocked in the Windows Firewall by default, unless LabVIEW has been granted access to communicate through the firewall in ”Windows Firewall settings”. To open firewall settings, first right click the windows button in the bottom left corner of the screen and select ”settings”. In the windows settings menu select ”Network and Internet” and select ”Windows Firewall”. In the Windows Firewall tab, select ”Allow an app or feature through Windows Firewall” from the menu on the left side. As shown in figure 3.13 scroll down in the ”Allowed apps and features” menu and check off all the boxes for ”LabVIEW (version) Development System”. If there are more than one version of LabVIEW installed on the computer, make sure the correct version is allowed to communicate through the Windows Firewall.



Allow apps to communicate through Windows Firewall

To add, change, or remove allowed apps and ports, click Change settings.

What are the risks of allowing an app to communicate?

For your security, some settings are managed by your system administrator.

Allowed apps and features:

Name	Domain	Private	Public	Group Policy
☐ iSCSI Service	☐	☐	☐	No
☑ JKI Updater	☑	☑	☑	No
☐ Key Management Service	☐	☐	☐	No
☑ LabVIEW 17.0 Development System	☑	☑	☑	No
☑ mDNS	☑	☑	☑	No
☐ Measurement & Automation Explorer	☑	☐	☐	No
☐ Media Center Extenders	☐	☐	☐	No
☑ Microsoft Bilder	☑	☑	☑	No
☑ Microsoft Content	☑	☑	☑	No
☑ Microsoft Edge	☑	☑	☑	No
☑ Microsoft family features	☑	☑	☑	No
☑ Microsoft Meldinger	☑	☑	☑	No

Figure 3.13: Allowing LabVIEW to communicate through Windows Firewall

Before the camera can be used to acquire images in LabVIEW for the first time, the driver setup from the accompanying Photron CD must be run. As the computer has no CD ROM drives, a copy in the attached software file can be used instead. Select ”SDK Setup 32” In the ”LabVIEW and MATLAB” folder and run the setup. This copies the driver library files (.DLL) to the appropriate windows folders on your hard drive. After completing the driver setup and if LabVIEW has access through Windows Firewall, you should be able to run the LabVIEW program to acquire the bubble images as described in chapter 4.3.

The camera is mounted on a wooden plate on the lift. There are four screws, one in each corner holding the wooden plate in place. These screws can be loosened to adjust the position of the plate. To move the handles back without tightening the screws, lift the handle straight up and move it around freely. For smaller adjustments of the camera position, there is an adjustment mechanism on the camera mounting which is shown in Figure 3.14. To fine-tune the camera position, the golden clamping knob have to come loose. Turn the clamping knob counter-clockwise to loosen it. While the clamp is loose, the camera position can be adjusted by turning the black knobs. Tighten the golden clamp knob after adjusting camera position to prevent the camera from sliding back and forth while running experiments.

When running the LabVIEW VI for the first time, lower the lift until the needle is visible in the camera feed to make sure the camera focus is correct. Instructions on how to access the camera feed and operate the lift are described in chapters 4.3 and 4.1 respectively. If the camera feed does not show the needle clearly, make sure the shutter speed is

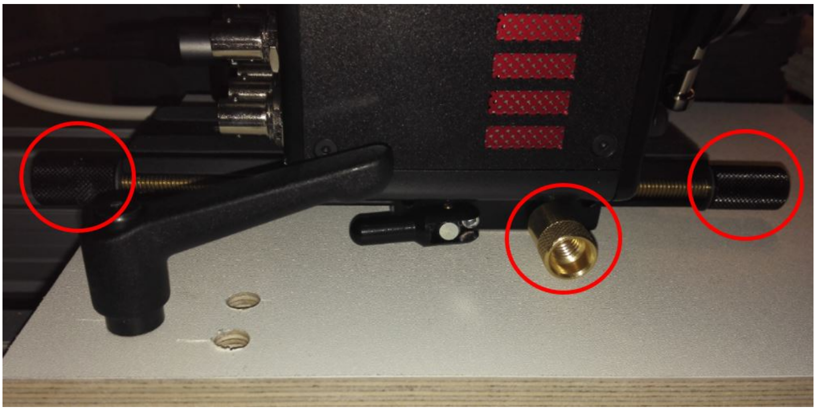


Figure 3.14: Mechanism for adjusting camera position

at an appropriate level (1/80'000sec recommended) and the lens is positioned 15.7cm from the column glass. When the camera is correctly positioned, adjust the focus by rotating the end of the camera lens until the needle image is as sharp as possible.

The background lighting source is a 150W Multiled QT manufactured by GSVitec®. The lamp is connected to the power supply with a CAT6 Ethernet cable. The lamp head is fitted with 12 LEDs to provide a more uniformly white background than a single light source would. To scatter the light even further, a "Rosco E+ 216" white diffusion-film is taped to the back of the column glass. A "Manfrotto 224 Variable Friction Arm" mounts the lights on the lift to make sure light is coming in from the same angle throughout the whole experiment.

3.6 Gas injection needle

Numbering (#) refers to component numbers from figure 3.15 and corresponding table 3.4. Glass needles used to inject gas bubbles are fastened to the column by the clamping of a rubber packing (2) when tightening the red plastic nut (3). The same principle is used in the Swagelok® Tube Fitting, which is a transition between the plastic tube and glass needle. To change needles, the top nut (4) needs to come loose for the rubber ferrule to let go of the needle. It is important to regularly check (eg. when changing needles) the ferrule (5) and the bottom of the needle for any damage or deformation. The rubber ferrule gets worn down from changing needles, which might harm the needle when tightened. Glass particles from a damaged needle might enter the needle and get stuck in the narrowing, which blocks the gas flow and prevents bubble injection. When fastening the top nut after changing needles, tighten the nut finger tight, before carefully tightening an additional $\sim$ 3/8 revolution. The recommended nut position is marked by a black permanent marker on the top nut and the Swagelok body. When the marker lines are aligned, the fitting should be tight enough. The difference between too loose fitting which results in gas leakage, and too tight fitting which breaks the needle is less than 1/2 revolution, so be careful when fastening the top nut. The bottom nut (8) clamps down on the steel ferrule (7) between the plastic tube and Swagelok body to provide a tight seal. There should not be a need to take off the bottom nut, but if it has to come off make sure to fasten it tightly, as this part is prone to gas leakage unless fastened properly. Try not to touch the Swagelok fitting or the plastic tube while the needle is attached to the column, as lateral movement may break the needle against the column base.

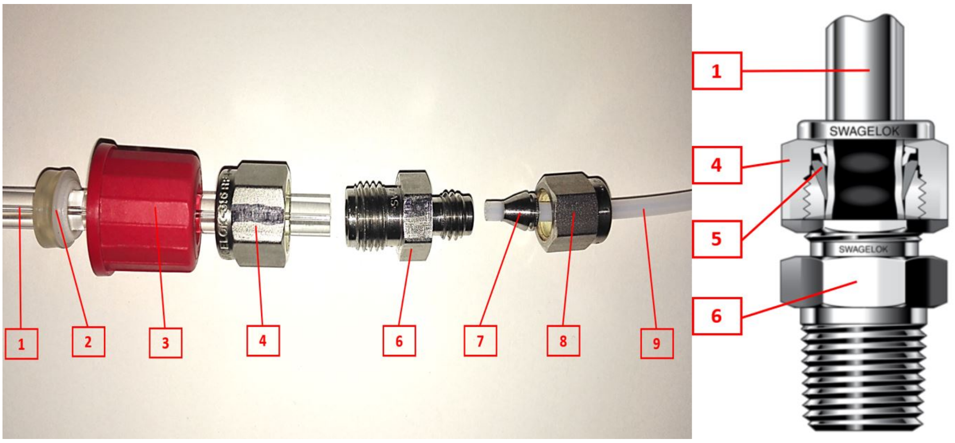


Figure 3.15: Swagelok® fitting components. Illustration (right side) from Swagelok company (2)

Table 3.4: Swagelok® fitting components

Number	Component
1.	Glass syringe
2.	Rubber packing
3.	Plastic nut
4.	Swagelok top nut
5.	Rubber ferrule
6.	Swagelok body
7.	Steel ferrule
8.	Swagelok bottom nut
9.	Plastic tube

Chapter 4

LabVIEW

LabVIEW is the software used to control all the hardware involved in injecting gas bubbles, moving the camera and acquiring images. The syringe pump and camera lift communicates through NI-VISA (National Instruments Virtual Instrument Software Architecture) programming interface, which enables the hardware to be controlled by text-string commands. In other words, VISA translates commands in LabVIEW to a language the controlling unit understands. Photron Fast Cam is not compatible with VISA and communicates through Photron's own drivers, hence the need to install "SDK setup" driver library before acquiring images in LabVIEW. Separate LabVIEW programs (Virtual Instruments "VI") has been made to control the syringe pump, camera lift and Photron Fast Cam separately, in addition to the main program for running experiments. Before attempting to run any of the LabVIEW programs, make sure all steps in the Hardware Setup chapter has been completed. All LabVIEW programs, SDK setup and drives are included in the attached software files. There should not be a need to make changes to the VI's block diagrams but if it should become necessary, look in appendix A.3 for a basic introduction on how to use LabVIEW.

4.1 Camera Lift

Being able to run the camera lift separately comes in handy when preparing to run experiments, as some needles are shorter than others. NB! Make sure all wires (Ethernet cables to camera, Multiled and the camera's power supply) are free to move along with the lift, as pulling wires too hard may break the equipment. Another thing to keep in mind is the way commands are carried out by the lift controller. Once an operation has started, the controller waits until the execution is complete before processing another command, which means you can not abort an operation in LabVIEW. To stop the lift while running press the "EMERGENCY STOP" knob on the controller. To start back up, rotate the "EMERGENCY STOP" knob clockwise until it pops back out and press the "POWER" button. The controller should go straight back to DNC-MODE when the power is back on. Operating the "Cameralift.vi" is fairly straight froward: Choose the velocity at which you want to run the lift and the relative displacement (how far you want the lift to go), choose direction and press the ”OK” button to run the lift while the VI is running. After the command is executed, velocity, displacement and run direction can be changed without stopping the program before pressing ”OK” again to carry out the command. The starting position of the lift (zero position) is set to wherever the lift was positioned when the power to the controller was turned on. Every time you stop the VI and start it back up by clicking the run arrow, the lift resets to the starting position. To change starting positions, first move the lift to the desired position, then restart the controller by using the ON/OFF button at the back of the controller. If the lift runs all the way up or down an end switch will be activated, which cuts the power to the lift. To release an end switch you have to turn the power back on and agree to release end switch on the controller interface manually. The weight carried by the lift appears to lower the lift position marginally, before the lift moves upwards. This will trigger the bottom end switch again after releasing, unless the lift position is adjusted by a manual operation. If the bottom and switch was triggered, use manual operation in the controller’s display to move the lift position a couple of millimetres up before running the VI again.

4.2 Syringe pump

When running the syringe pump via LabVIEW you need to know which Communications Port is assigned the USB connection. Instructions on how to find the correct Communications Port are described in chapter 3.3. The rest of the configuration is similar to a manual configuration in the syringe pump display. In addition to selecting flow rate, pump direction and bubble volume, you can also choose how many bubbles to inject and the time delay between the bubbles. When running the VI it completes one cycle, injecting the chosen number of bubbles before stopping. To run another cycle press the run arrow again. If the pusher block hits a stopper collar or any other error appears on the syringe pump’s display, stop the vi by clicking the stop button in the top menu. To clear the error, press the yellow error message that appears on the pump’s touch screen interface. Typical values for a bubble injection are flow rates from 3 - 20 mL/hr and the bubble volumes from 500 nano litres to 2 micro litres. The parameter values depends on needle size.

4.3 Photron Fast Cam Mini AX100

The program used to acquire images is a modified version of Photron’s own sample VI. Some minor modifications have been made to allow scrolling and re-sizing of the front panel while running the program as well as the option to force the VI to stop. These modifications makes the front panel easier to navigate, while the program is still the same as the ”SDK Sample” VI provided by Photron. If the program should become corrupted or deleted, it is easily recreated from the ”SDK Sample” VI by clicking ”File” from the top menu, and selecting ”VI properties” to open the VI properties tab. Open the ”Window Appearance” tab from the category menu and select ”Default”. This will allow scrolling,

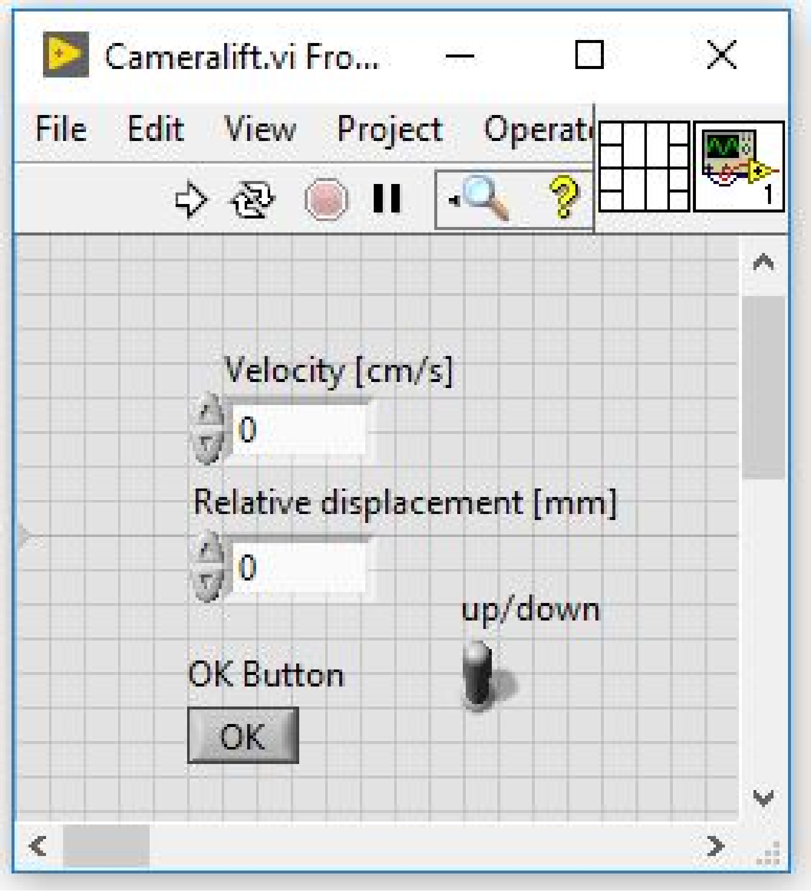
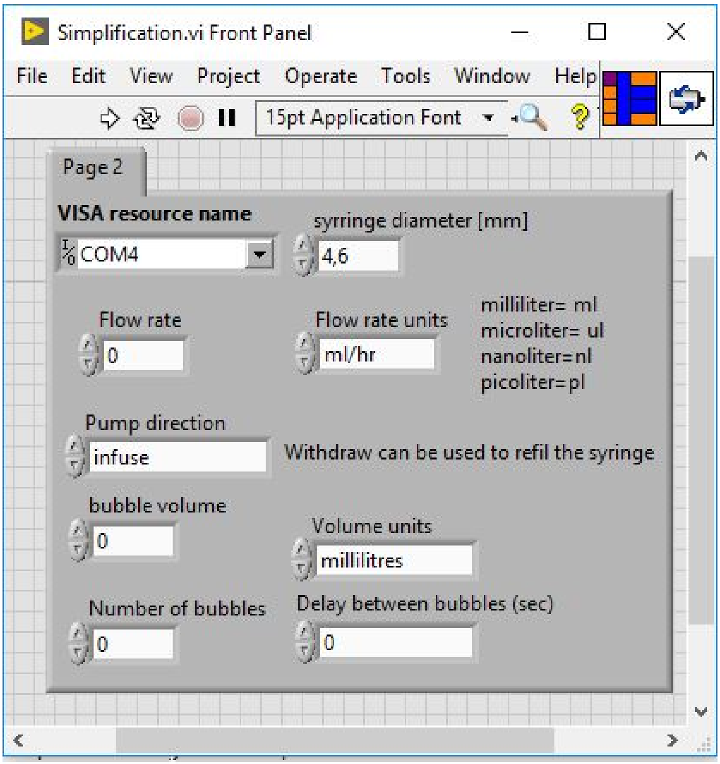


Figure 4.1: Lift control front panel

re-sizing and display the tool bar while running.

When running the VI, the program will ask for the Connection type and IP Address. Select ”Gigabit Ethernet” connection, ”Detect Auto” and enter the IP Address: 192.168.0.xxx if it does not appear on its own. Shutter speed is usually the only setting which needs changing before recording images, since the default setting is proportional to selected frame rate. In this setup a shutter speed of 1/60’000 sec or faster is required due to the intensity of the background lighting. In some cases the different trigger modes could also be useful. The camera keeps an image buffer of ∼21’000 frames in its internal memory, depending on the resolution. The buffer grants access to images recorded before a recording is triggered. The trigger mode ”START” saves all images between a recording is triggered and stopped or the internal memory runs out. ”CENTER” trigger mode also saves all images between a recording is triggered and stopped or the internal memory runs out, and an equal number of images from before the recording is triggered, which means the number of images from



VISA resource name: COM4

syrringe diameter [mm]: 4,6

Flow rate: 0

Flow rate units: ml/hr

milliliter= ml
microliter= ul
nanoliter=nl
picoliter=pl

Pump direction: infuse

Withdraw can be used to refil the syringe

bubble volume: 0

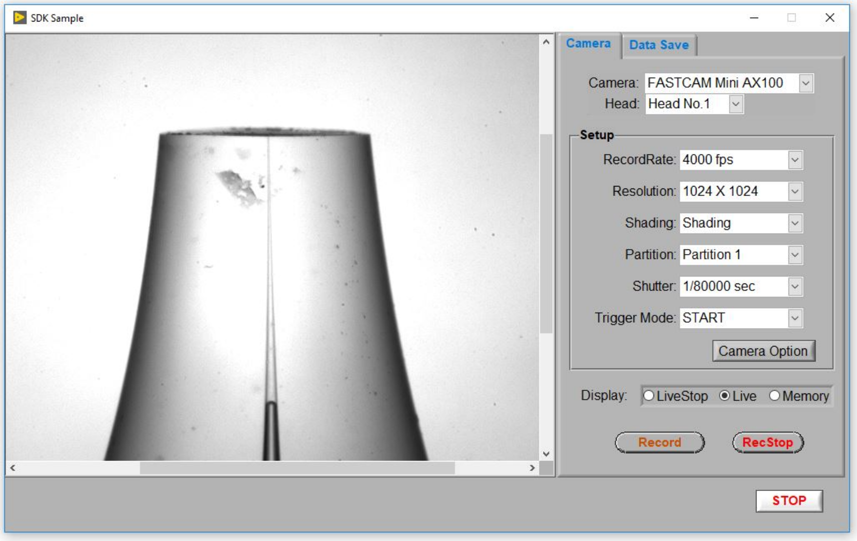
Volume units: millilitres

Number of bubbles: 0

Delay between bubbles (sec): 0

Figure 4.2: Syringe pump control panel

a recording is triggered and the internal memory runs out is halved. ”END” trigger mode saves the last image buffer in internal memory from before a recording is triggered. ”RANDOM” trigger mode randomly selects the ”starting point” of the recording in the image buffer. To record images, activate record mode by clicking ”Record”, then click ”Waiting TRG” when ready to start a recording. Click ”Rec Stop” to stop recording. Downloading images to the computer is done in the ”Data Save” tab, where you can review the recorded images and select which ones to download. Choose where to save the recording by clicking the folder next to ”Save Path” and type in a recording name in ”File Name”. To download several parts of the recording separately (eg. image 1-200 and 4500-6000), go back to the ”Camera” tab after downloading the first part, then reopen the ”Data Save” tab to download the next part.

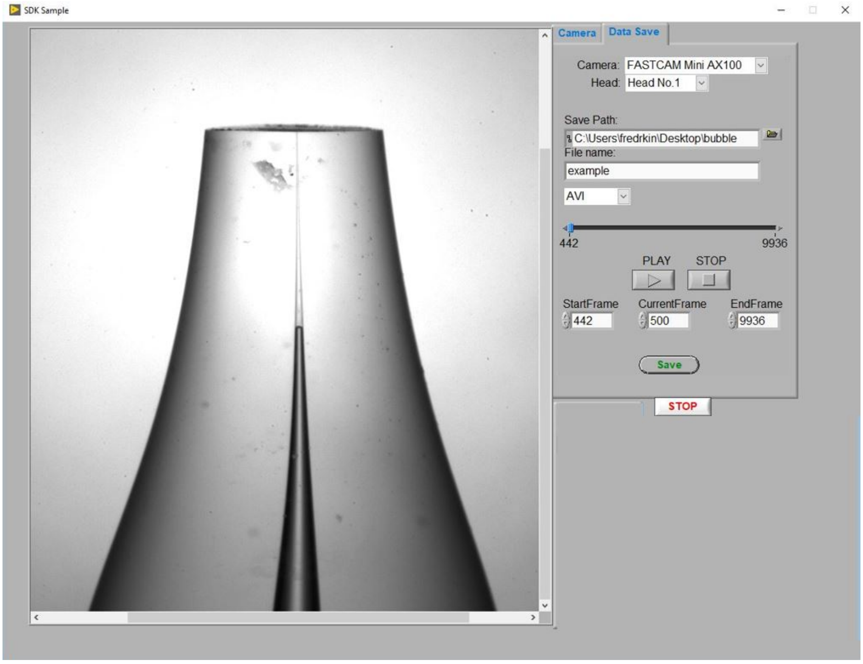


Camera: FASTCAM Mini AX100
Head: Head No.1

RecordRate: 4000 fps
Resolution: 1024 X 1024
Shading: Shading
Partition: Partition 1
Shutter: 1/80000 sec
Trigger Mode: START

Display: ○ LiveStop ● Live ○ Memory

Figure 4.3: Photron Fast Cam control panel ”camera” tab



Camera: FASTCAM Mini AX100
Head: Head No.1

Save Path: C:\Users\fredrkin\Desktop\bubble
File name: example
AVI

442 9936

StartFrame 442
CurrentFrame 500
EndFrame 9936

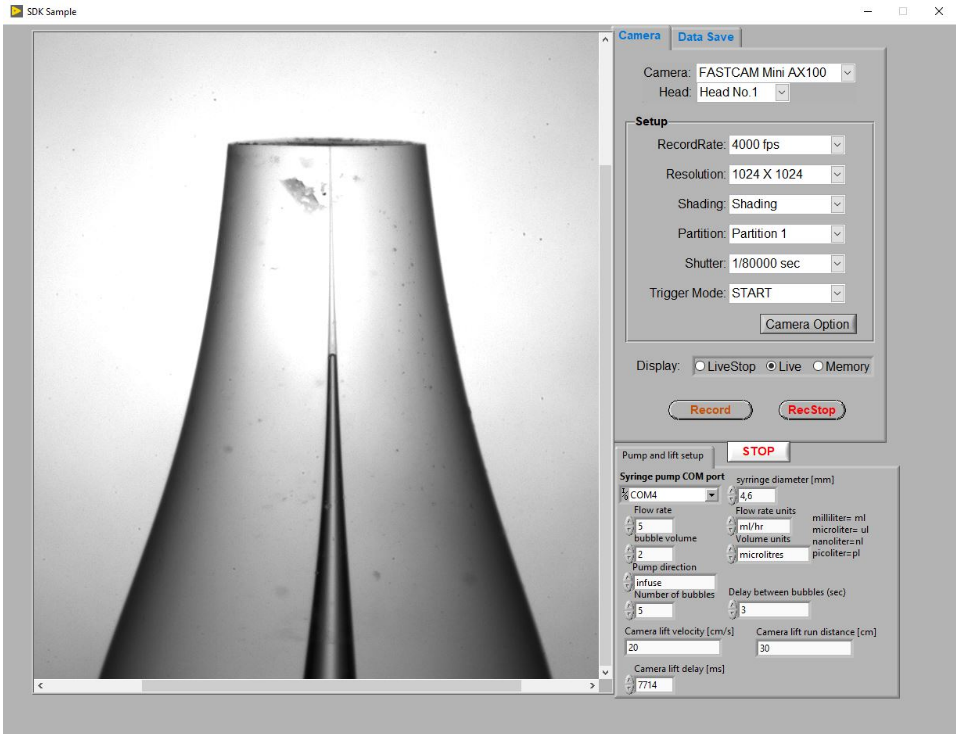
Figure 4.4: Photron Fast Cam control panel ”data save” tab

4.4 Recording rising bubbles

The program for recording rising bubbles is a combination of the previously mentioned VIs. Camera lift, syringe pump and image acquisition are all run in parallel to reduce execution time. As described in section 6.2.2 the delay from an injection command is sent to the bubbles are produced is predictable, which is what we use to trigger the camera movement.

Before running the VI, configure the syringe pump and camera lift setup as described in the previous sections. A new feature in the setup is the camera lift delay, which is set to correspond with the time it takes from an injection command is sent (recording started) to the bubble is formed subtracting the time it takes for the lift controller to process and execute the run command (65 ms) . After completing the syringe pump and camera lift setup, run the VI and configure camera settings as described in section 4.3. When running the VI, the syringe pump configuration commands are sent immediately. If the syringe pump is not correctly configured before the VI runs, the ”VISA session” might not close, which prevents the ”STOP” button in the front panel to stop the VI. In this case the VI needs to be stopped by using the ”Abort Execution” button in the tool bar

To run the experiment click ”Record” and ”Waiting TRG”. The status change from ”Waiting TRG” to ”Recording” triggers the bubble injection command and starts the countdown on the camera lift delay. After a set delay the run command is sent to the lift controller. Keeping the time delay to a minimum is advised as the recording starts immediately when triggering the recording, and the internal memory lasts about 5,3 seconds at a frame rate of 4’000fps. When all bubbles are injected and the lift stops, click ”Rec Stop” and look in the ”Data Save” tab to see the results. VISA communication to the syringe pump is closed after completing each cycle, which makes it necessary to stop the VI by clicking the ”STOP” button and running the VI again to perform a new recording. If the VI is not restarted before starting another recording, the camera and lift will proceed as usual, but no bubbles will be produced. Pressing the ”STOP” button before the syringe pump has completed a cycle will cause an error which requires you to stop the VI by using the ”Abort Execution” button in the tool bar, or force LabVIEW to close by using task manager (ctr+alt+del). As LabVIEW runs the syringe pump VISA session in parallel to the image acquisition, and the ”STOP” function is coded into the drivers, the image acquisition loop needs to be the last one to close. Closing the image acquisition loop disables all controls inside it, which means you are not able to stop the VI correctly and store image data if the bubble injection sequence is not closed. Stopping the VI while the ”recording” state is active prevents you from accessing the recorded images as the ”Rec stop” state activates the collection of image buffer data. As long as the ”Rec stop” was activated before stopping the VI, image data from the recording can be accessed and saved when running the VI again.



Camera: FASTCAM Mini AX100

Head: Head No.1

Setup

RecordRate: 4000 fps

Resolution: 1024 X 1024

Shading: Shading

Partition: Partition 1

Shutter: 1/80000 sec

Trigger Mode: START

Display: ○ LiveStop ● Live ○ Memory

Syringe pump COM port: COM4

syrringe diameter [mm]: 4,6

Flow rate: 5

Flow rate units: ml/hr

bubble volume: 2

Volume units: microlitres

milliliter= ml
microliter= ul
nanoliter=nl
picoliter=pl

Pump direction: infuse

Number of bubbles: 5

Delay between bubbles (sec): 3

Camera lift velocity [cm/s]: 20

Camera lift run distance [cm]: 30

Camera lift delay [ms]: 7714

Figure 4.5: Program for recording rising bubbles

Chapter 5

Image analysis

Image analysis is done in MATLAB, due to difficulties processing AVI (Audio Video Interleave) files in LabVIEW. The image analysis is performed on uncompressed AVI videos of the experiments. The reason for performing the image analysis on a single AVI video instead of downloading the experiment data as a series of separate images in another format (eg. BMP) is because of practicalities. There are no benefits or downsides to the quality of the image analysis, but it is more practical to only have a single video for each experiment instead of downloading up to 21'000 images. The image analysis of an AVI file in MATLAB is also faster than the processing of separate BMP images, as it takes time to open each file.

When using the MATLAB script to perform image analysis, make sure the correct framerate and lift velocity is selected to accurately estimate the bubble position in every frame. When downloading the recording of the experiment in LabVIEW as explained in section 4.3, it is recommended to select the frame where the lift starts moving (or the bubble departs from the needle, which might be easier to see) as the "startframe". If you want to include the formation of the bubble in the video, note down at which frame this occur relative to the chosen "startframe", and enter the frame number as "movestart" in the MATLAB script to get an accurate bubble position. To chose which video to process the name of the video file must be entered in the VideoReader function (eg. VideoReader('filename.avi')). Also make sure MATLAB is looking for the video in the correct folder. If a different zoom than maximum zoom is selected on the camera, the pixel size needs to be re-calibrated, and entered as 'Ps'. At full zoom the pixel size is calibrated to 87,5 [$\mu m^2/pix$]. The calibration was done by measuring the width of the teknolab needle in μm and then measuring its width in pixels from an image of the needle mounted inside the column. For the bubble mass calculation the specific gas constant 'R', water temperature 'T' and water level 'h' needs to be accurate. Note that water level 'h' is measured from the tip of the needle, and not the bottom of the column.

When running the MATLAB script, it opens the video selected in the VideoReader function, and reading the frames one by one in a while loop. The while loop runs until there are no more frames to read from the video. Inside the loop, the frames are then converted to binary, giving all bright pixels the value 1 and dark pixels the value 0. The conversion to a binary image (imbinarize) is done with an adaptive threshold, where the threshold for a pixel's brightness depends on the brightness of its surrounding area. After converting the image to binary, a morphological operation is added to fill the bright spot in the middle of the bubble, before counting the number of pixels in the bubble (number of pixels with the value 0). The bubble radius (assumed spherical) is then calculated and logged in the first column of an array 'Rb'. The estimated bubble position is calculated using the time and lift velocity and logged in the second column of the "Rb" array. As mass transfer is the object of interest, the bubble mass is estimated and logged in the third column of the 'Rb' array. To estimate bubble mass the pressure is calculated by using the position of the bubble, which is then used to estimate gas density.

After the MATLAB script has completed the image analysis, the bubble radius and position can be read from the "Rb" array. The bubble position in the second column shows how far the bubble has travelled in centimetres, starting with zero at the tip of the needle (in reality zero is where the lift starts running, which should be at the tip of the needle). The estimated bubble mass can be read from the third column in the "Rb" array. Keep in mind that these values are estimated with the assumption of a spherical bubble, which is not accurate for larger oscillating bubbles. The matlab script is included in the attached software.

4pt

```
clear all;
%-------------------------------------------
%adjust these values to correspond with the settings in labVIEW.
framerate=4000; %video framerate [fps].
vL=20; %lift velocity [cm/s].
movestart=0; %Frame at which the lift starts moving.
Ps=87.5; % pixel size [um^2 /pix] from callibration.
v = VideoReader('following.avi'); %read video file. Replace 'filename.avi'.
%for calculation of bubble mass
R=287.5; %Specific gas constant [J/(kg*K)]
T=293.15; %Temperature [K]
h=200; %water level [cm]
rho_l=997; %liquid density [kg/m^3]
%press run
%-------------------------------------------
i=0; % i used for counting frames
j=0; % j used for indexing bubble size array

while hasFrame(v)
    video = readFrame(v); %read pixel grayscale values (1024x1024 matrix) for each frame
    BW = imbinarize(video,'adaptive','ForegroundPolarity','bright','Sensitivity',1); %convert to binary
    BW2 = bwareaopen(BW,100000); %remove/fill bright spot in the middle of bubble
    Bs=sum(BW2(:) == 0);%count number of pixels in the bubble (pixels with the value '0')
    i=i+1; %counting frame number
     time= (i-movestart)/framerate; %how long the lift has been moving
     while time >= 0
        j=j+1; %indexing bubble size, position and mass in array
    Rb(j,1)=sqrt(Bs*Ps/pi);% convert bubble area in pixels to bubble radius in microns (assumed spherical)
    %imshow(BW2);%show the processed image
    position= (vL/1000)*time/100; %position of lift (and bubble) in [m], where the starting position is is 0.
    Rb(j,2)=position;
    p= 101325 + rho_l*9.81*(h-position*100); %Pressure = 1 atm + rho_l * g * z [pa]
    rho_g= p/(R*T); % density of gass bubble [kg/m^3]
    mb= (rho_g*((Rb(j,1)*10^(-6))^3)*4*pi/3); %Bubble mass= rho_g * Vb [kg]
    Rb(j,3)=mb;
    time=-1; %Set time less than 0 to stop while loop from executing again on the same videoframe
     end
end
% Rb array (bubble radius, position, mass)
```

Chapter 6

Experiment

The experiments done with the rig were tests on which bubble sizes could be produced, and which parameter settings that could produce controlled single bubbles. Different needle designs and materials were tested, and attempts to record the rising bubbles were made, using both an analysis of the live camera feed and by using a time delay to start the camera movement.

6.1 Bubble production

The first experiments were done with a needle from Teknolab made from a hydrophobic ”PEEK silicone” material with an inner diameter of 75µm, and a glass needle with an inner diameter of approximately $1\mathrm{mm}$. The designs of the needles were also different. The silicone needle was a 15 cm long capillary tube with a uniform inner diameter, whereas the glass needle had an inner diameter of 1 mm, with a narrowing down to 100µm at the tip. The tip of the glass needle was unfortunately broken off before any tests were conducted.

Running experiments with the needle from Teknolab proved difficult, because initially when the gas injection started nothing happened. When the injecting syringe was almost empty, all the gas came out at once, producing up to a hundred bubbles over a short period of time. This was likely caused by the design of the needle. As the tip of the needle is a 15cm long section with an inner diameter of 75µm, the pressure drop was probably too high for a reliable bubble injection. The volume of the tube $\mathrm{V_t}$ is much larger than critical volume $\mathrm{V_{crit}}$, which might have allowed too much air compression before overcoming the required pressure at the tip of the needle to produce a bubble. The PEEKSil needle is not transparent, which makes it hard to conclude whether water intrusion played a role in causing this effect.

The needle from Technolab also had a problem with bubbles sticking to the tip of the needle while forming. Despite the relatively small inner diameter of the needle, the bubbles generated were much larger than intended. As shown in fig.6.1 the bubbles appears to stick to the needle until buoyancy forces are strong enough to break the surface tension between the bubble and needle tip.

Injecting gas through the glass needle proved to be more reliable and generated smaller bubbles, despite having an inner diameter over ten times larger than the needle from Teknolab. The glass needle has a relatively wide inner diameter with a narrowing at the end. By reducing the length of the narrow section of the needle, friction forces are greatly reduced. Bubbles did not stick to the glass needle either, which indicates both the design and material in the glass needles are better suited for bubble injection than the silicone needle.

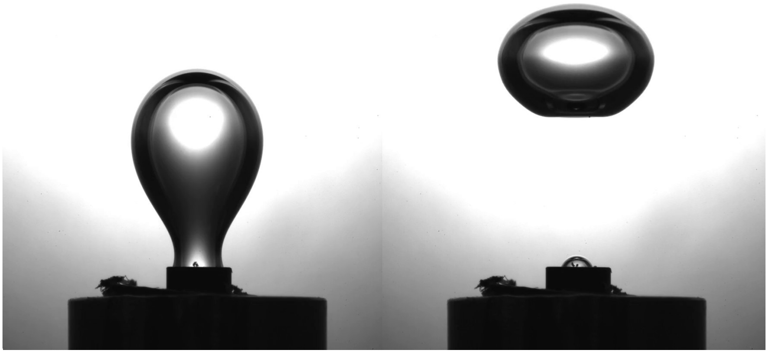


Figure 6.1: Bubble injection with needle from Technolab

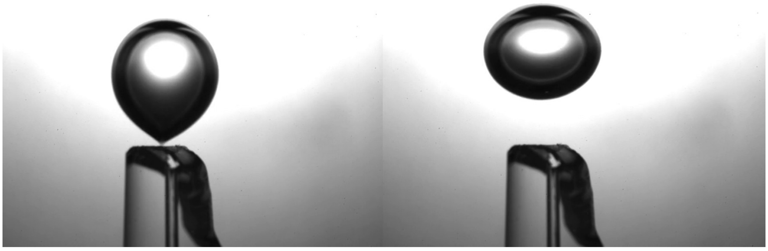


Figure 6.2: Bubble injection with glass needle

After reviewing the results from these tests, new glass needles were ordered with an inner diameter from 100 µm down to 10 µm.

With the new glass needles a reliable production of bubbles with a diameter less than 2 mm was achieved. These needles had an inner diameter of 0.8 mm, with a narrowing of the inner diameter at the tip of the needle. The new glass needles had an inner diameter ranging from 10-90 um at the needle tip. The minimum bubble size was achieved by a needle with a diameter of 20 μm. The smallest achieved bubbles had a diameter of ∼ 0.6 mm. Producing bubbles with needle diameters smaller than 20 μm proved to be difficult, as the ”ketchupbottle effect” was dominant for these needles. Figure 6.3 and 6.4 shows a series of images of a bubble departing from the 20 μm and 15 μm needles respectively.

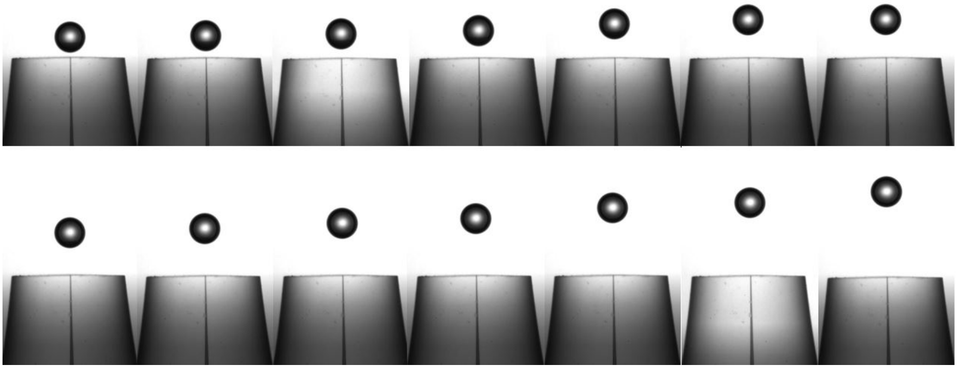


Figure 6.3: Bubble departure from 20 μm needle

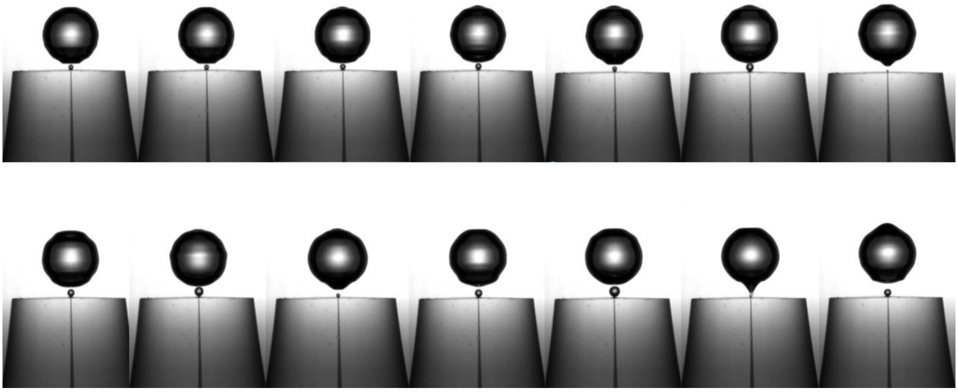


Figure 6.4: Bubble departure from 15 μm needle

Figure 6.4 illustrates the ”ketchupbottle” effect. Previous to the production of the bubble shown in the image a volume of 500 nL was injected by the syringe at a rate of 5 ml/h three times with a waiting period of 30 seconds between each injection. No bubbles were produced from the first three injections. On the fourth injection the bubble in figure 6.4 was produced almost instantaneously. As shown in the image series the formation of new bubbles are faster than the acceleration of the detached bubble, which results in the forming bubbles merging with the departed bubble, increasing its size and preventing it from accelerating. In addition to the increased bubble size due to the absorption of forming bubbles, the bubble size was much larger than expected upon the initial departure from the needle. One of the needles were accidentally broken while switching needles, which caused small glass particles to remain in the Swagelok fitting. These glass particles then got lodged inside the 15 µm and 20 µm needles while running experiments, rendering them unable to produce bubbles. Due to a misunderstanding when ordering new needles, the needles that were purchased to replace the broken ones had an inner diameter of ∼1 mm before the narrowing. The new needles also had a different design, where the narrow section at the tip of the needles were significantly longer.

Bubble injection with the new needles proved difficult, as the ”ketchupbottle effect” was dominant, even for 20 µm needles who gave the most promising results with the previous design. The 20 µm needle with the longest narrow section was unable to produce bubbles at a water depth of 2 m. When injecting the whole 1 mL syringe, this needle produced no bubbles at all, but as the water level decreased while draining the column, bursts of bubbles were ejected from the needle. The pressure drop over the narrow section of the needle was probably too large for the conditions of forming a bubble from equation 2.5 to be met. The 20 µm needles with a shorter narrow section were able to produce bubbles at 2 m depth, but the timing was unreliable and the ”ketchupbottle effect” caused the bubble size to be much larger than intended. Figure 6.7 and the accompanying table 6.1 compares the needle properties of the currently available needles (b and c) to the needle which gave the best results (a).

Table 6.1: Needle properties

Needle	Diameter [mm]	Length of narrow section	Bubble injection
a	0.8	Short	Controlled single bubble
b	∼1	Long	Uncontrolled bursts
c	∼1	Very long	No bubbles

The needles that were able to produce a controlled single bubble had diameters of 20 um and 80 um. The bubbles where injected with volumetric flow rates from 3 - 20 mL/h, which is bellow the critical flow rate. The resulting bubble sizes shown in figures 6.6 and 6.7, appears to confirm that the bubble size is independent of flow rates bellow critical flow rate. Graph data is presented in appendix A.1. The results from the 20 um needle shows a trend where bubble size decrease with increasing flow rates, but this is likely due to a low sample base.

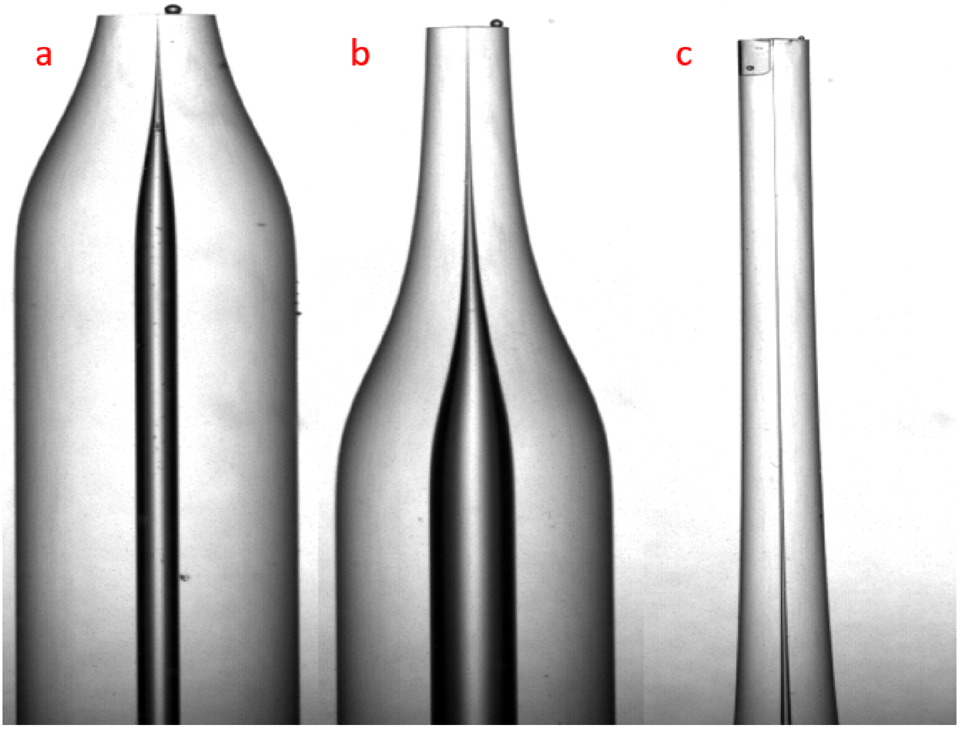


Figure 6.5: 20 µm needles with different designs.

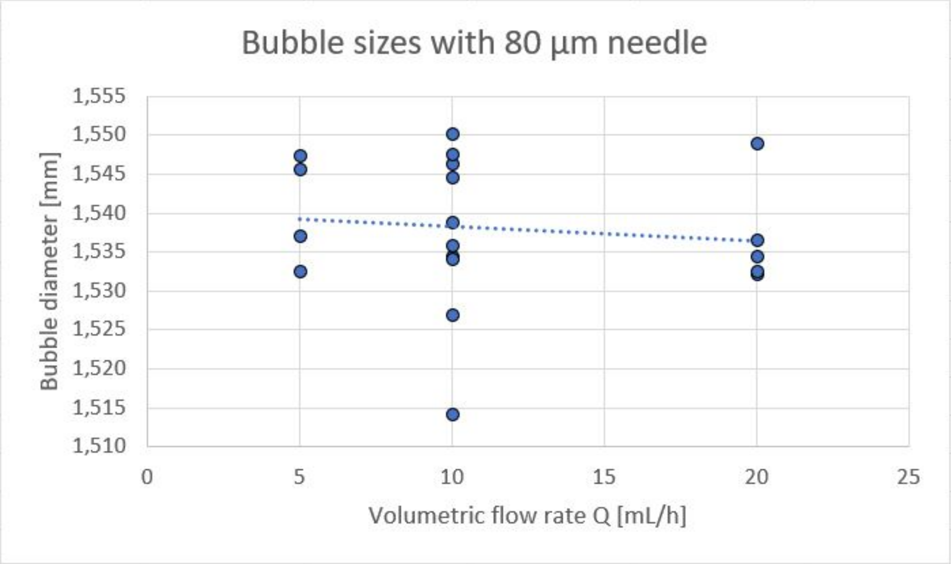


Figure 6.6: bubble sizes produced with a 80 µm needle.

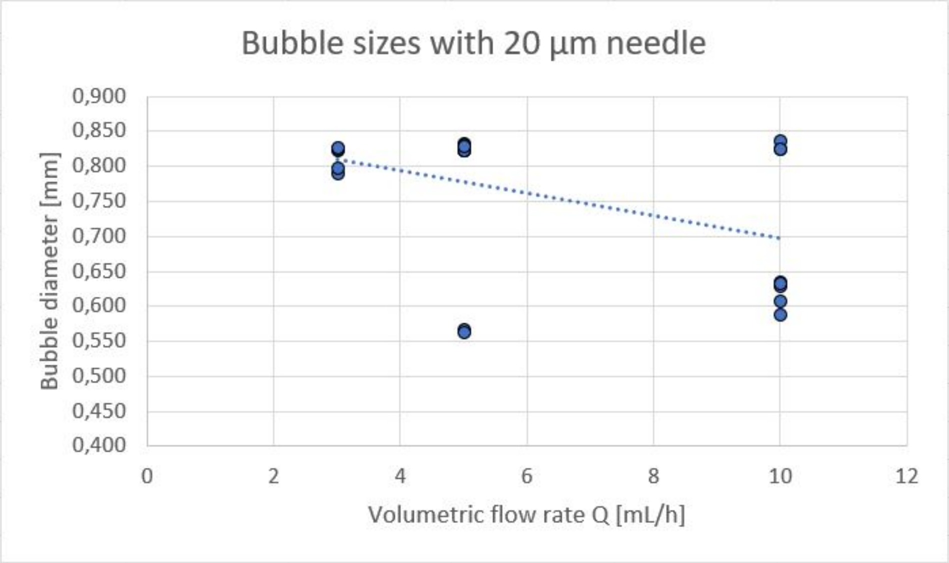


Figure 6.7: bubble sizes produced with a 20 µm needle.

6.2 Image acquisition

The initial plan to acquire images of the rising bubbles was to use LabVIEW's image processing of a live camera feed to recognize bubbles, which would trigger the camera and use the bubble position to adjust the speed of the lift. This method proved to be far too slow due to a slow image acquisition, and the fact that the lift briefly stops when changing the velocity. The next approach was to find a correlation between the time it takes from an injection command is sent, to the bubble is produced. By using the time delay as a trigger for the lift and camera, the reliability of the image acquisition improved greatly.

6.2.1 Image processing of camera feed

The image processing of the camera feed converted the image from an 8-bit image into a binary image, where all pixels with a grayscale brightness value above threshold was assigned the value 0, whereas pixels containing darker objects were assigned the value 1. After the image was converted to a binary image, three filters were added to remove border objects, small particles and to fill holes as the corners of the image have a slightly lower light intensity and some small ”dots” remains after removing border objects. In the middle of the bubble the light intensity is as bright as the background, hence the need for the ”Fill Holes” function, which converts all pixels inside a circle of pixels with the value 1 to the same value. After the image was processed the ”Image Analysis” function was used to recognize whether there were particles in the image or not. When particles are present, the ”Image analysis” block returns the number of particles, which is used to trigger the lift and camera. Unfortunately as the acquisition of data from the camera and transforming it into an image takes 150 ms, and the image processing takes about 24 ms. By the time the camera and lift were triggered, the bubble had already passed.

Removing the conversion from raw data to an image format improved the acquisition speed to 50ms. As the LabVIEW image processing do not process raw data, another method for recognizing bubbles was necessary. The raw data acquired from the camera is a one dimensional array with one element for each pixel (1024^2), where the element value is the grayscale brightness of the pixel (0-255). Since the background is uniformly white the value of every pixel is $\sim$255. By using a threshold for the minimum value in the raw data array as a trigger, the LabVIEW script can recognize bubbles and trigger the camera and lift at a rate of 20fps. The drawback of using this method, is that any particle that reduces the brightness value of one or more pixels below threshold will ”pull the trigger”. This approach also proved to be too slow, as there is a slight delay between the ”trigger is pulled” and the lift starts to move. This delay is caused by the fact that the run command is sent to the lift on the next iteration of the loop which adds $\sim$50 ms and the time it takes for the controller to receive and process the command which is 65 ms.

6.2.2 Timed trigger

As image processing failed to provide a fast enough response, the next method used in attempt to record the rising bubbles was a timed trigger. By predicting the time it will take from a bubble injection command is sent to the bubble is formed, the time delay can be used as a trigger to move the camera lift. A prerequisite for this method to work is a consistent time delay between the command is sent to the bubble is formed.

To estimate the injection time, tests were run where the time between an injection command was sent to the bubble departed from the needle were measured. The results from the initial tests are shown in figure 6.8 and 6.9. As can be seen in the two graphs showing the injection times for a 1 mm and 80 um needle, the variation in injection time was very high. In order to use injection times to trigger the camera recording and lift the variation needs to be greatly reduced. At full zoom the height of the camera image covers about 1 cm, which gives a tolerated variation in injection time of ±0.5 cm divided by the bubble velocity. As the tolerance increases inversely with bubble velocity the tolerated variation in injection time will be different for each needle size. Needles which produces bubbles with a diameter of $\sim$ 1.5 mm will have the lowest tolerated variation of $\sim$14 ms in pure water, while the smaller bubbles have a tolerated variation up to 100 ms.

Injection time mainly depends on the three following variables: Water intrusion inside the needle, VI's iteration speed and gas injection rate. The VI's iteration speed is close to constant and the gas injection rate is controlled, which leaves water intrusion as the main concern. The water intrusion seems to depend on the three variables water depth, needle design and time. As water depth and needle design does not vary, injecting bubbles at a constant time interval should keep all the variables for injection time close to constant, which should reduce its variation.

A series of test were run to measure the injection time with a controlled delay of three and two seconds between each bubble with the 80 um and 20 um respectively. The tests were run in series of five bubbles per test. The time delay between each test run was not controlled or measured, since some time was needed between tests to browse through the video and download the images. Comparing figures 6.8 and 6.9 to figure 6.10 and 6.11 confirms the previous assumption that controlling the time between bubbles are produced reduces variation in injection time. Table 6.2 shows the average injection time and standard deviation for the three graphs, excluding the first bubble in each series (sample 1,6,11 etc.) from figure 6.10 and 6.11 since the time delay between each series is not controlled. Sample 7 from figure 6.10 and sample 10 from figure 6.11 are also excluded from the calculations for table 6.2, since these bubbles were abnormalities, which does not represent a controlled bubble production. Unfortunately the 20 um needle was broken after running the first two series of tests. Due to the low number of samples with this needle, the statistics presented in table 6.2 might be inaccurate.

Table 6.2: Bubble injection data

Needle	Average injection time [ms]	Standard deviation [ms]
Randomly timed injection.		
1 mm	625	301.1
80 um	473	173.3
Timed injection with a controlled delay between each injection.		
80 um	658	37.0
20 um	490	89.7

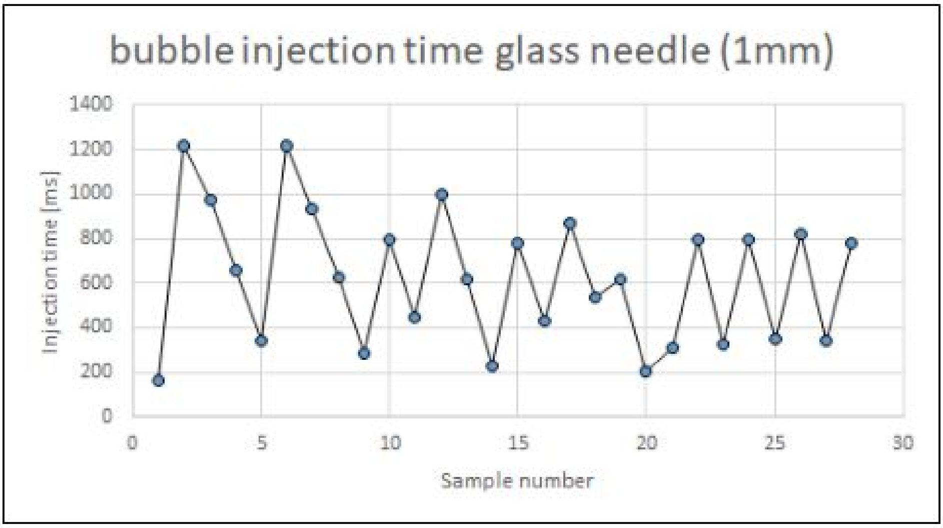


Figure 6.8: Randomly timed bubble injection with glass needle d∼1 mm

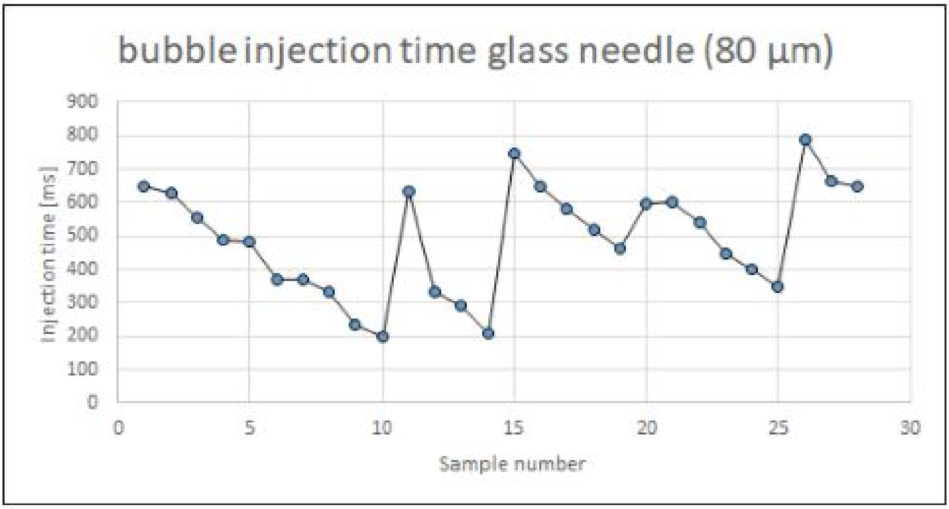


Figure 6.9: Randomly timed bubble injection with glass needle d=80 µm

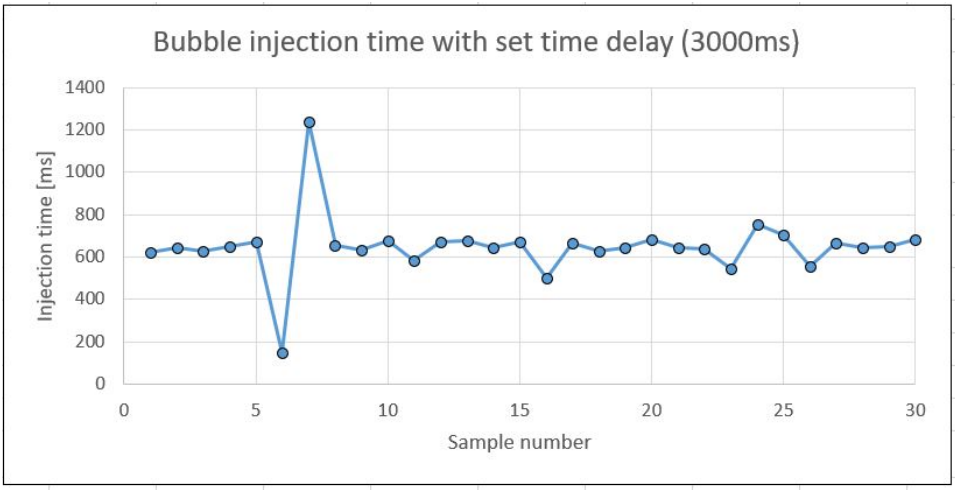


Figure 6.10: Bubble injection with glass needle d=80 µm, where six series of five bubbles were introduced with a time delay of 3000 ms between each bubble in the series.

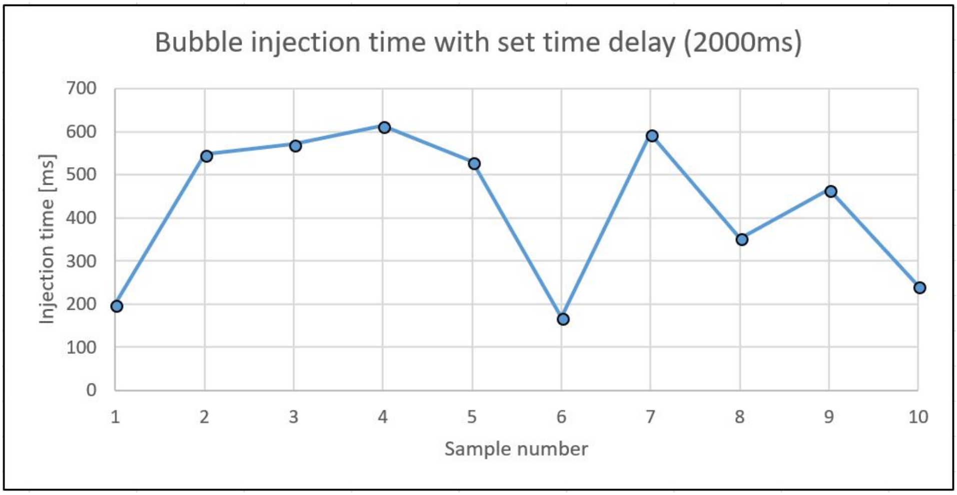


Figure 6.11: Bubble injection with glass needle d=20 µm, where two series of five bubbles were introduced with a time delay of 2000 ms between each bubble in the series.

6.2.3 Recording rising bubbles

Knowing the time delay before a bubble is formed and the bubble velocity, allows the lift to accurately follow the bubble. As all the needles capable of producing bubbles with a diameter less than 1 mm were either broken or unsuited for running tests due to the ”ketchupbottle effect” causing large variations in injection times, the attempts to record rising bubbles we performed with a 80 μm needle. The bubbles produced by the 80 μm needle had a diameter of ∼1.6 mm. In accord with figure 2.2 observations showed that bubbles with a diameter higher than ∼0.9 mm has a terminal velocity in pure water which is higher than the lift's maximum velocity. Figure 6.12 shows a series of images from an attempt to follow a rising bubble made with a 80 μm needle. The bubble passes the camera, which is moving at the maximum velocity of 25 cm/s, with an estimated velocity of ∼36 cm/s.

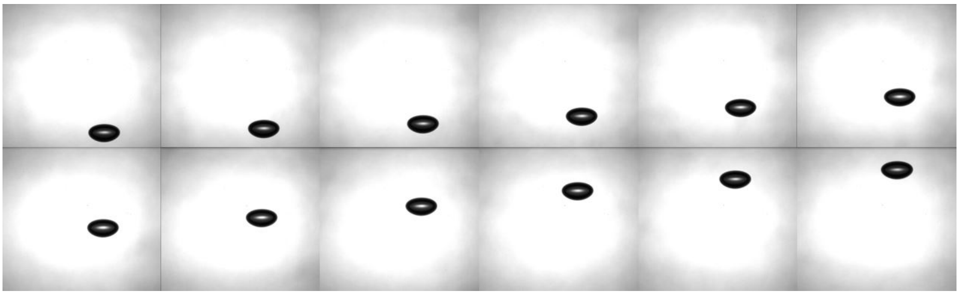


Figure 6.12: Bubble rising past the camera, which is moving at maximum velocity.

An attempt was made to slow down the terminal velocity of these bubbles, by using tap water. The idea was that the tap water would be sufficiently contaminated to reduce terminal velocity to less than 25 cm/s. Alas the Trondheim drinking water was too pure, and the reduction in terminal velocity was minor. In an attempt to increase the velocity of the lift, as the bubbles were not slowed down sufficiently by using tap water, the current and maximum velocity of the lift were increased to beyond its intended capacity. This resulted in the lift being able to achieve velocities up to ∼40 cm/s, but only for a short distance (∼30 cm) before overloading. An order of a new linear unit with a ballscrew pitch of 10 mm is in the works, but will not be completed before the due date of this report. The new linear unit will be able to achieve a velocity of 50 cm/s, which should be more than enough to follow bubbles of all sizes. New needles with inner diameters lower than 80 um has been ordered, but will not arrive before this report is submitted. Without the ability to run the lift faster than 25 cm/s or produce smaller bubbles, recording gas bubbles rising through water along the full length of the column is not possible at this time.

Chapter 7

Analysis

7.1 Bubble production

Observations on the bubble injection quality of the different needle designs that were tested appears to be in line with the implications of equations 2.4 and 2.6. As the chamber volume $\mathrm{V_t}$ is higher than $\mathrm{V_{crit}}$ the compression of gas causes instantaneous gas flow rate through the tip of the needle Q to surpass the flow rate at which gas is injected by the syringe $\mathrm{Q_{in}}$, similar to a mechanical spring which is released once the compression reaches a critical level. A needle design with a long narrow section at the tip causes a significant pressure drop, which increases the required chamber pressure. This seems to enhance the ”ketchupbolle effect” significantly. This type of needle design has a very large variation in injection time, and produces bubbles much larger than the theoretical minimum $\mathrm{V_F}$. The large bubble size is caused by a high volumetric flow rate larger than $\mathrm{Q_{crit}}$due to compression in the chamber volume, and as shown in figure 6.4, the formation of new bubbles is faster than the acceleration of detached bubbles, causing them to merge and increase the bubble volume even further.

The best needle design for injecting very small single bubbles were needles with an inner diameter of 0.8 mm before the narrowing at the tip, with the shortest possible narrow section. An inner diameter of 0.8 mm appears to be sufficiently small not to allow any significant gas compression, while being large enough not to cause pressure drop problems. The short narrow section creates minimal pressure drop, which allows for a low pressure in the chamber volume when producing bubbles. Producing bubbles at low chamber pressure appears to yield the most controlled injections, due to a low degree of gas compression, which reduces the instantaneous gas flow rate Q and lowers the variation in injection time.

7.2 Image acquisition

The best way to acquire recordings of the rising bubbles appears to be the timing of bubble injections, where the injection time is used to trigger the camera movement. All attempts to use the live image feed to control the camera movement had too long response times. The results from measuring the injection time of bubbles produced with an 80 um needle had 43,5 % of the produced bubbles within the tolerance of 14.5 ms (tolerance calculated for bubble diameter of 1.6 mm which has a terminal velocity of 34.57 cm/s). The 20 um needle had a much higher variation in injection time with 12.5 % of the produced bubbles within tolerance. Parameter values used for calculating terminal velocity and tolerated variation in injection time is presented in appendix A.2. As the sample base for the 20 um needle is very small, these results should only be interpreted as an indication of which results to expect, and not an accurate representation of the real injection time. As the needles are hand made, some variation in the design is also expected, which is likely to give different results for the new 20 um needle when it arrives. However, the results clearly show that with the current setup, one can not expect the camera movement to hit the mark every time. The increased variation in injection time for smaller needles is likely caused by the increased resistance due to a narrower inner diameter, which causes compression of gas in the chamber volume. Reducing the chamber volume would probably reduce the variation in injection time for smaller needles, and possibly also allow a controlled bubble injection with needles bellow 20 um. Another way to increase the chance to successfully acquire a recording of the rising bubble is to increase the tolerated variation by zooming out. Zooming out should only be used as a last resort since it will reduce the accuracy of the image analysis. More specifically the resolution of the bubble decreases since image pixels covers a larger area, which will increase the uncertainty at the borders of the bubble.

The current setup was unfortunately not able to follow rising bubbles along the full length of the column, which prevented the evaluation of whether velocity change due to pressure drop along the column is a problem or not. Analysis on mass transfer in bubbles with a diameter less than 2 mm could not be performed either, due to the fact that the lift was not able to match the bubble velocity. Using liquids with a higher viscosity than water should reduce the bubble velocity, but due to a lack of time this was not tested. It should be noted that if this approach is to be tested in the future, liquids that are easily washed away should be chosen since cleaning the inside of the column will require a lot of time and work.

Chapter 8

Conclusion

The results from bubble injection tests showed that the main challenge to a reliable production of very small single bubbles is preventing compression of gas in the chamber volume. The key factors for preventing gas compression are needle design and chamber volume. Keeping the pressure drop over the needle and chamber volume as low as possible is crucial in achieving the smallest bubble sizes. Preventing gas compression also appears to be a key factor to reducing variation in injection time, but the amount of data to back up this assumption is insufficient at this time. The lower limit on achievable bubble size is ~0.6 mm with the current setup, which was achieved with a 20 um needle. A new tube between the syringe and needle with a lower inner diameter has been ordered. A reduction in chamber volume with the new tube might allow production of even smaller bubbles, since it should allow the use of smaller needles.

After countless attempts on using real time image analysis of the live image feed from the camera, the conclusion on how to trigger camera movement is that timing the bubbles is the best option. This approach is not perfect, as the variation in injection time makes the camera movement ”miss the mark” sometimes, but it is still the better option. The lift should be able to follow bubbles with a diameter less than 0.92 mm, but this has not been tested since the small needles were broken. Larger bubbles has too high terminal velocities for the lift to follow. When the new linear unit arrives this should no longer be a problem as its maximum velocity is 50 cm/s. Another possibly lift related problem is the fact that the pressure drop over the column will cause the bubbles to expand, which increases the terminal velocity of the bubble. This could cause problems with following bubbles with a diameter from 0.5 - 1 mm where change in bubble size affects the terminal velocity the most. In cases with mass transfer this could actually be an advantage, as the reduction in bubble size due to mass transfer might be balanced by the expansion caused by the pressure drop. However, this is just speculation as no tests were actually performed.

Bibliography

[1] H. N. Oguz and A. Prosperetti, “Dynamics of bubble growth and detachment from a needle,” vol. 255, 1993.

[2] H. A. Jakobsen, *Chemical Reactor Modeling, Multiphase Reactive Flows*, 2nd ed. Springer, 2008.

[3] I. GUWAHATI, “Mass transfer operations,” 2012.

[4] I. K. Kure, “Numerical study of mass transfer limitations in a bioreactor operated in the heterogeneous flow regime,” 2017.

[5] P.-C. Chiang and S.-Y. Pan, *Application of Mass Transfer Models in Environmental Engineering*, 1st ed. Intech, 2015.

[6] C.-J. Huang and C.-H. Kuo, “General mathematical model for mass transfer accompanied by chemical reaction,” vol. 9(2), 1963.

[7] N. Kraakman, J. Rocha-Rios, and M. Loosdrecht, “Review of mass transfer aspects for biological gas treatment,” vol. 91(4), 2011.

[8] P. M. Doran, *Bioprocess Engineering Principles*, 2nd ed. Elsevier Ltd., 2003.

[9] J. Vejrazka, M. Fujasova, P. Stanovsky, M. C. Rizicka, and J. Drahos, “Bubbling controlled by needle movement,” vol. 40(7-8), 2007.

[10] S. Baz-Rodriguez, A. Aguilar-Corona, and A. Soria, “Rising velocity for singe bubbles in pure liquids,” vol. 11, 2012.

Appendix

A.1 Bubble size tables

Table 1: Bubble sizes with 80 um needle

80 um **needle**		
Volumetric flow rate [mL/h]	**Cross section area [pixels]**	**Equivalent bubble radius [mm]**
20	21194	1,537
20	21136	1,535
20	21073	1,532
20	21537	1,549
20	21085	1,533
10	20580	1,514
10	21464	1,546
10	21415	1,545
10	21569	1,550
10	21135	1,534
10	21172	1,536
10	21255	1,539
10	21128	1,534
10	20931	1,527
10	21496	1,548
5	21210	1,537
5	21493	1,547
5	21447	1,546
5	21082	1,533
5	21308	1,541

Table 2: Bubble sizes with 20 um needle

20 um **needle**		
Volumetric flow rate [mL/h]	**Cross section area [pixels]**	**Equivalent bubble radius [mm]**
10	6103	0,828
10	6074	0,824
10	6171	0,834
10	2898	0,824
10	2855	0,830
10	6126	0,825
10	6297	0,823
10	6104	0,829
5	6076	0,823
5	5610	0,791
5	5720	0,798
5	6075	0,823
5	6125	0,826
5	3624	0,635
5	3555	0,629
5	3324	0,609
5	3592	0,633
5	3120	0,590
3	6148	0,828
3	6100	0,824
3	6244	0,834
3	6100	0,824
3	6189	0,830

A.2 Terminal velocity and tolerated variation calculations

$$\mathrm{V_T} = \frac{1}{(\frac{1}{\mathrm{V_{T1}}}\frac{1}{\mathrm{V_{T2}}})^{1/2}} \tag{1}$$

$$\mathrm{V_{T1}} = \mathrm{V_{T,pot}}[1 + 0.73667\frac{(\mathrm{gd_e})^{1/2}}{\mathrm{V_{T,pot}}}]^{1/2} \tag{2}$$

$$\mathrm{V_{T,pot}} = \frac{\Delta\rho \mathrm{gd_e^2}}{36\mu_\mathrm{l}} \tag{3}$$

$$\mathrm{V_{T2}} = (\frac{3\sigma}{\rho_\mathrm{l}\mathrm{d_e}} + \frac{\mathrm{gd_e}\Delta\rho}{2\rho_\mathrm{l}})^{1/2} \tag{4}$$

$\mu_\mathrm{l} = 0.89 \cdot 10^{-3}\ \mathrm{Pa \cdot s}\ ,\ \rho = 72.86 \cdot 10^{-3}\ \mathrm{N/m}\ ,\ \Delta\rho = (997.0 - 1.4)\ \mathrm{kg/m^3}\ ,\ \mathrm{g} = 9.81\mathrm{m/s^2}$
$\mathrm{d_{e(80\mu mneedle)}} = 1.6 \cdot 10^{-3}\mathrm{m}\ ,\ \mathrm{d_{e(20\mu mneedle)}} = 6.0 \cdot 10^{-4}\mathrm{m}$

$\mathrm{V_{T((80\mu mneedle)}} = 34.57\mathrm{cm/s}$

$\mathrm{V_{T((20\mu mneedle)}} = 13.18\mathrm{cm/s}$

Tolerated variance:

80 µm needle $= \frac{0.5cm}{34.57cm/s} = 14.5$ ms

20 µm needle= $\frac{0.5cm}{34.57cm/s} = 37.9$ ms

A.3 Basic LabVIEW tutorial

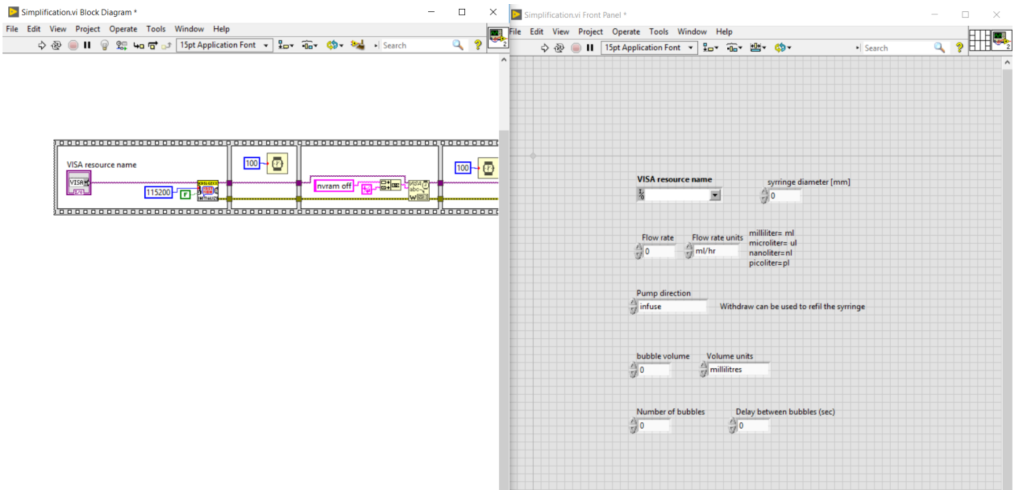
Basic LabVIEW tutorial

Introduction to LabVIEW

LabVIEW is a program development system from National Instruments. The software has a broad spectrum of applications, with a reasonably simple and intuitive user interface. The main difference between LabVIEW and conventional programming software is the use of graphical function blocks instead of text based code. In this tutorial you will learn the basics required to understand and operate the bubble generator and object tracking instruments, using LabVIEW.

Virtual Instrument (.VI)

A Virtual Instrument (.VI) is the format of the program you write in LabVIEW. All VIs has a block diagram, which is where you connect the function- and control- blocks to perform different tasks, and a front panel, which is the interface you use when operating the controls and running the programme. In other words, the block diagram determines which tasks the VI will perform, and the front panel is a virtual representation of your instrument panel, hence the name Virtual Instrument.


Simplification.vi Block Diagram *
File Edit View Project Operate Tools Window Help
15pt Application Font
Search
VISA resource name
VISA
115200
F
100
nvram off
100
Simplification.vi Front Panel *
File Edit View Project Operate Tools Window Help
15pt Application Font
Search

VISA resource name

syringe diameter [mm] 0

Flow rate 0

Flow rate units ml/hr

milliliter= ml
microliter= ul
nanoliter=nl
picoliter=pl

Pump direction infuse

Withdraw can be used to refil the syringe

bubble volume 0

Volume units millilitres

Number of bubbles 0

Delay between bubbles (sec) 0

Figure 1: Screenshot of Block Diagram (to the left), and the Front Panel interface (to the right).

When you open a VI, only the front panel is displayed. In order to show the block diagram, open the <<Window>> menu at the top of the front panel and choose <<Show block diagram (ctrl +E) >> or <<Tile left and right (ctrl + T) >> to show both panels next to each other.

To run the VI you can use the run button to run the VI once, or run continuously to run the “script” several times until stopped. Usually loops are a better option to running the VI continuously. To stop or pause the VI you can use the stop and hold buttons next to the run buttons. In the block diagram there are some additional options in the top menu. Highlight execution (the light bulb) highlights the data flow as the VI is running, which can be a usefull tool when troubleshooting. If the run arrow looks broken, there is an error preventing the program from running. Clicking the broken arrow shows the errors preventing the VI from running.

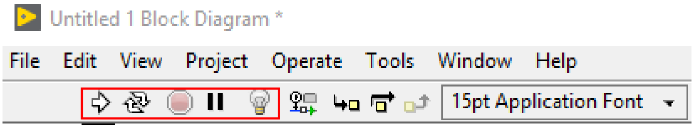


Figure 2: Buttons for running and stopping a VI and highlighting data flow.

Functions- and controls palate

There are two different palates from which we get our building blocks. In the front panel we use the <<Controls palate>> to choose which type of controls and indicators we want to use in the instrument panel. The <<Functions palate>> is used to choose the operations performed in the block diagram.

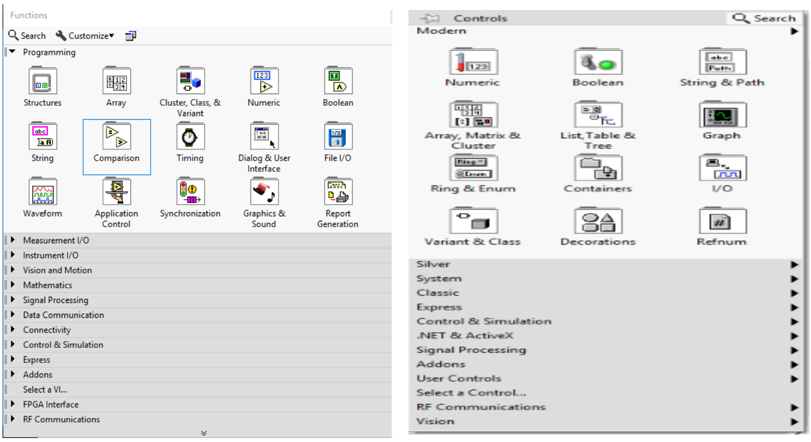


Figure 3: Functions palate (to the left) and controls palate (to the right)

If you are unsure about what a function- or control block does, you can right click it and select the <<help>> option, to open the LabVIEW help section for that particular block. Controls are shown in both front panel and block diagram, unlike functions who only appear in the block diagram. Double clicking a control in one of the windows will show the same control in the block diagram.

The three main signal formats in the functions are numeric signals, Boolean signals (true/false) and texts strings. Below you see the three different signal controllers, with their respective indicators. The numeric signals appear orange like the figure below when it is a floating number (decimals) and blue for whole numbers (integers).

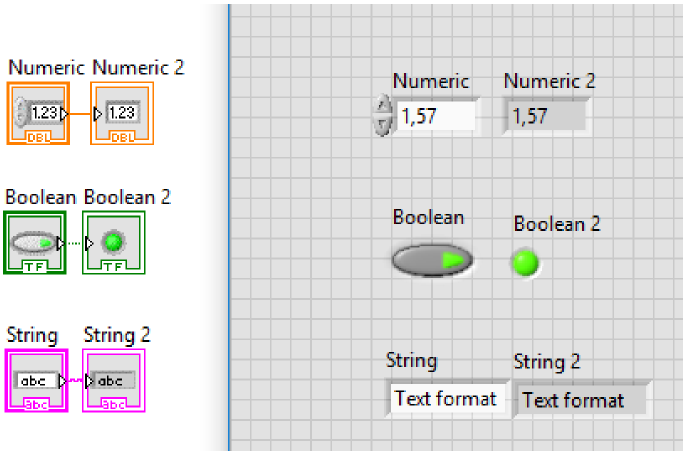


Figure 4: Controllers and indicators. Block diagram to the left and front panel to the right.

There are also enumeric controls, which are text options with an assigned numeric value. Choosing one of the options in the front panel control, will make the function block take on the specified numeric value. This is especially handy when choosing indexes. If you write all the commands in a single text string on different lines, you can use the index number of the command you want to execute. Below there is an example on how to choose the command using an enumeric constant. The number display in the front panel shows the index value of the colour we want to choose.

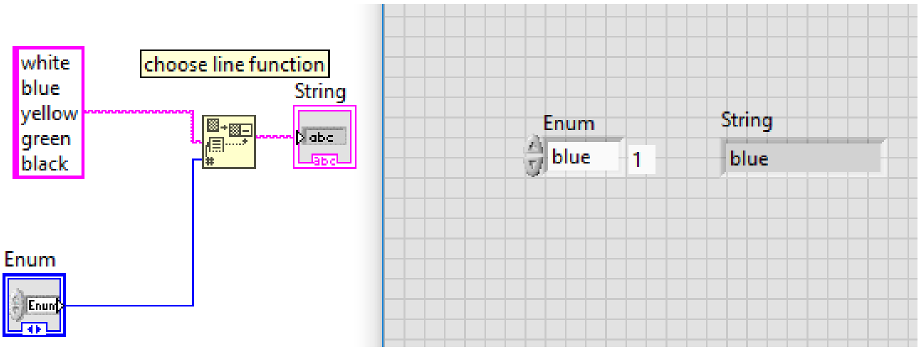
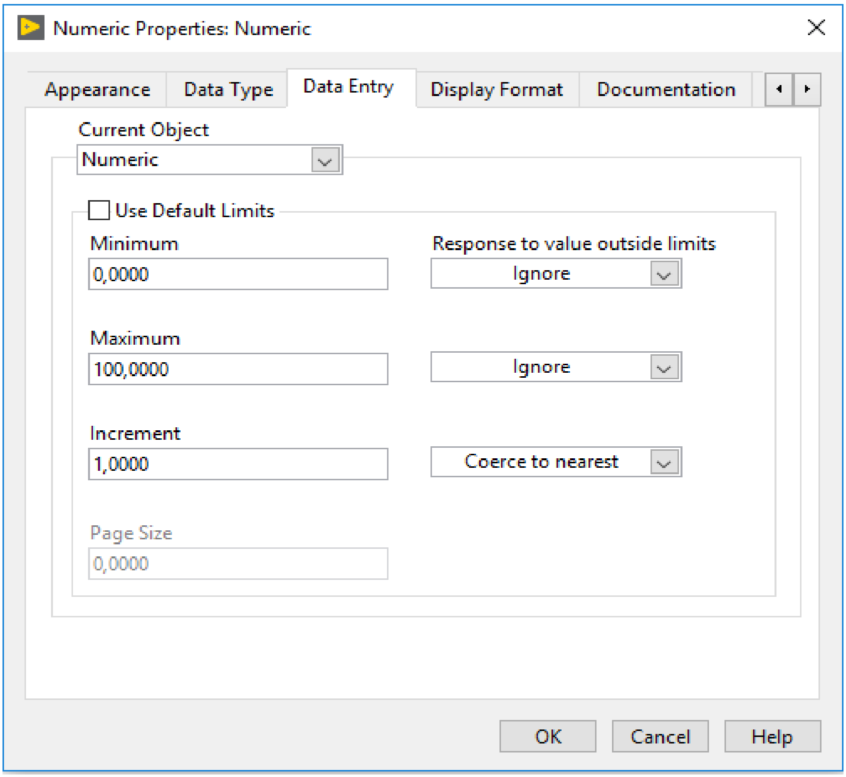


Figure 5: Enumeric control example

By right clicking a control or indicator in the front panel and selecting <<properties>>, you can change the appearance, data type, data entry (numeric limits and increment size for controls), display format (digits of precision/ significant digits), and some additional features which are not covered in this tutorial.



Current Object: Numeric

☐ Use Default Limits

Minimum: 0,0000 — Response to value outside limits: Ignore

Maximum: 100,0000 — Ignore

Increment: 1,0000 — Coerce to nearest

Page Size: 0,0000

Figure 6: Property window.

Property nodes

Properties can also be read/written by using property nodes. By right clicking the controller/indicator in the block diagram, select <<create>>, then select <<property node>> and the property you want to read/write. The property node block with the same name as your controller/indicator will now appear.

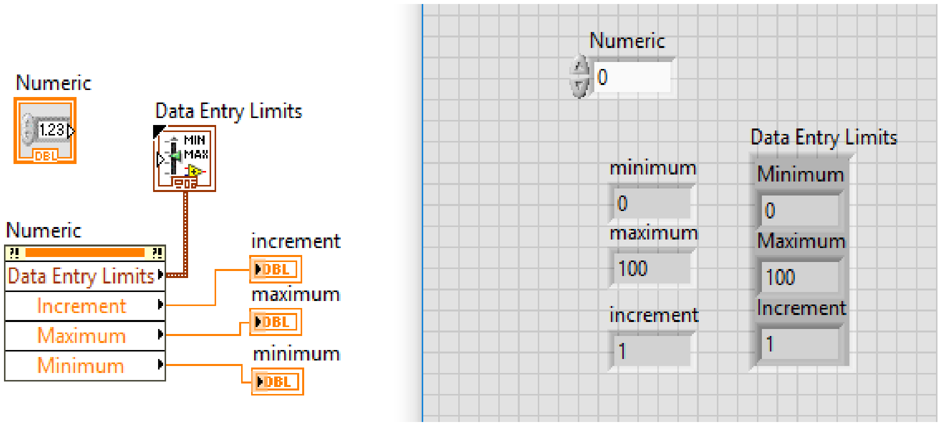


Figure 7: Example on use of property nodes.

The properties can be read individually or as a cluster with all the data entry limits for this case. By right clicking the property node and selecting <<change (all) to write>>, you can use the property node to specify the data entry limits.

Local variables

The values of controllers and indicators are often used for several different purposes, which can lead to messy block diagrams with a lot of wiring. To avoid messy block diagrams, we use local variables. By right clicking a controller/indicator, selecting <<create>> and <<local variable>> you create a local variable in the block diagram. The local variable takes on the value of its controller without any wiring connecting them. Note that if the local variable is placed outside the controller's loop, data will not be passed from the controller to the local variable before the loop has stopped.

Key functions

Timing

The timing palate contains functions which keeps track of time, and time delay functions. The <<wait function>> is one of the more frequently used functions used in the bubble production VI. By assigning a numeric value to the block input, the function will keep the data flow from continuing for numeric value in milliseconds .

Structures

In the structures palate we find the loop and sequence functions. Unless you want the functions to execute only once, they need to be inside a loop. If you want your commands and functions to execute in a specific order, the sequence structure is most likely the solution to your problem.

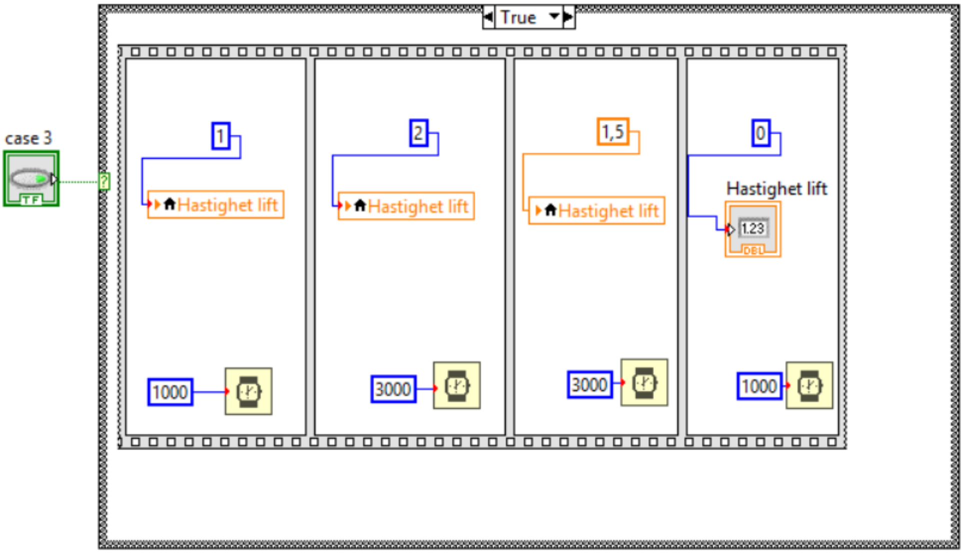


Figure 8: Case structure with a flat sequence

In the figure above, you see an example of how you can use the “case-structure” and the “flat-sequence”. The case structure initiates the loop shown in the picture if the condition is true. When the controller (named *case 3* in this example) has a false value, the loop in the ‘False’ window is run.

The sequence within the case structure, will run in a loop as long as the condition is true. The sequence in this example is used to control the velocity of a lift in a predetermined sequence. The wait function (the clock), determines how long each part of the sequence last in milliseconds. The velocity is in this case given by constant numbers for each step in the sequence.

There are also other loops, like the ‘while loop’, which executes the loop until a stop condition is met. And the ‘for loop’, which executes a set number of loops. These functions are similar to loops and sequence functions in other programming languages.

Passing data from inside a structure to the outside is done by regular use of wires. A tunnel is created at the loop boundary. By clicking the tunnel, you can select the tunnel mode, which is either concatenating (pass on data for every loop iteration), last value (only data from last iteration is passed on) or indexing (data is stored for every iteration, and passed on as an array of values after the loop is stopped).

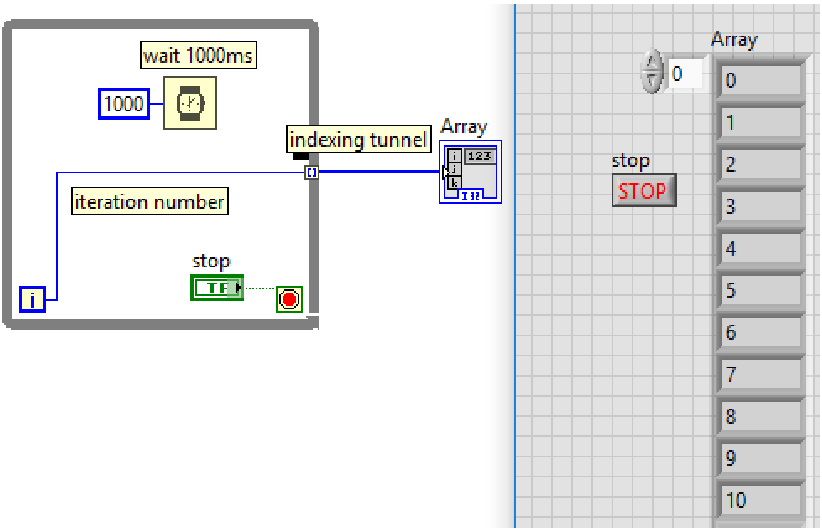
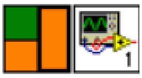


Figure 9: While loop with indexing tunnel

The while loop runs a new iteration every second, and storing the iteration number until the loop is stopped. When the loop is stopped, an array with all iteration numbers is passed through the tunnel.

Sub VI

For tasks that requires many complex functions, it is common to create a separate VI for each function. These VIs are used as “sub-VIs” who are called upon by the main VI to easier keep track of the main functions and minimizing the number of functions in the block diagram.



This figure is in the top-right corner of your front panel. Here you choose how the function block, which calls to your current VI, will look. Each rectangle is connected to the control/indicator you want to write/read. By using this function block in another VI, it is used as a sub-vi, which runs in the background and executing its functions. Using the example above as a sub-vi. The main VI would provide a numeric input to determine the index value, and would read the text string command. This removes the need for having the command list constant, and the chose line function in the block diagram. To open the sub-VI. You only need to double click the function block in the block diagram, and the front panel appears (make sure the main program is not running). By exploring the sub-VIs, you get an understanding of the deeper levels of the functions hierarchy. Sub-VIs can also call on other VIs, making several levels of VIs executing functions, which eventually ends up providing the outputs you use to execute this experiment.

VISA

Virtual Instrument System Architecture (VISA) is a standard I/O API, used by many of the major instrumentation companies like Hewlett-Packard, National Instruments and Tektronix.

VISA can control GPIB, serial, USB, Ethernet, PXI, or VXI instruments, making the appropriate driver calls depending on the type of instrument you use so you do not have to learn instrument-specific communication protocol. It is platform-, bus- and environment- independent. In other words, the same API is used regardless of device type, platform, or programming language.

GPIB, serial, USB, Ethernet, and some VXI instruments use message-based communication. You program message-based instruments using string type commands. The instrument has a local processor that parses the command strings and sets the appropriate register bits to perform the operations you want. The Standard Commands for Programmable Instruments (SCPI) standardizes the command strings used to program compliant instruments. Similar instruments often use similar commands. Instead of learning different command messages for each type of instrument from each manufacturer, you need to learn only one command set. The message-based functions used for the syringe pump are VISA Open, VISA Read, VISA Write, VISA Set I/O Buffer Size and VISA Close. We also use property nodes to set baud rate and timeout. All the functions are found under <Instrument I/O> <VISA> in the functions palate.

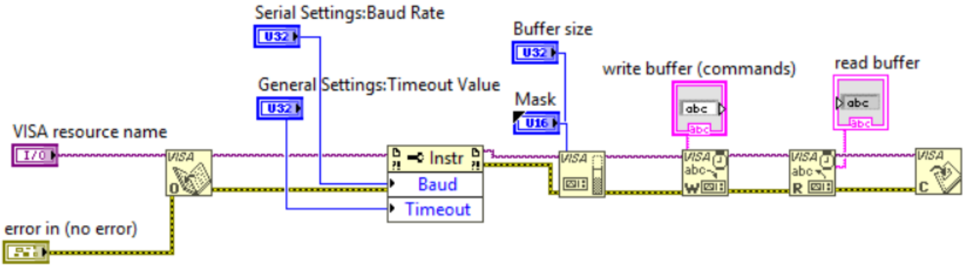


Figure 10: Example VI for executing a command and reading status using VISA for USB device.

First, we start the session with the VISA Open function. Baud rate and timeout is set using a property node. Some devices may require you to specify the size of the I/O buffer. This is done by specifying the number of bits in VISA Set I/O Buffer Size function. Mask specifies which buffer you want to set <I/O Transmit>, <I/O Receive> or both. Now that you have set baud rate, timeout and buffer size, you can write commands to your instrument using VISA Write and read its status with VISA Read. To close the session use VISA Close. You might want to add an error handler to the VISA Close function to read possible errors.