HeinSight – Liquid Level GUI

User documentation

About

HeinSight is a computer-vision based control system for automating common laboratory tasks. The Liquid-Level Hydrogen-1 release is a prototype designed to facilitate the remote monitoring and control of liquid-air interface in a transparent vessel across a variety of pre-programmed experimental types that require continuous stirring:

1) Single and dual pump continuous preferential crystallization (CPC)
2) Continuous distillation
3) Filtration

The GUI, described in this documentation, allows users to set-up and run the system without having to interact with code. All necessary files can be downloaded at: https://gitlab.com/heingroup/heinsight_liquid_level_H1

Documentation related to the code can be found in docstrings and comments included in each script.

Hardware Requirements

The system is currently programmed to run using one or two NE-9000 peristaltic pumps (http://www.syringepump.com/peristaltic.php) and a Logitech C922x Pro Stream Webcam (https://www.logitech.com/en-ca/product/c922-pro-stream-webcam).

Other hardware may be used but the code will need to be modified accordingly.

Software Requirements

Running the program requires the following software:

- PyCharm or other IDE - https://www.jetbrains.com/pycharm/
    - Note: In development and usage we used the PyCharm Community Edition
- Python3 - https://www.python.org/
    - We have tested that the code is compatible with Python versions: 3.6.5 - 3.7.3
- Logitech Camera Settings (or other webcam control software) - https://support.logi.com/hc/en-us/articles/360024695174-Downloads-C920s-HD-Pro-Webcam
- *HeinSight* Liquid-Level scripts - https://gitlab.com/heingroup/heinsight_liquid_level_H1

Typographic Conventions Used in the Documentation

The documentation uses different fonts for particular content types:

- **Arial 11 bold red** is used for the links to software demo videos
- **Arial 11pp bold** is used for actions
- `Courier New 11pt` is used for file names and folder names
- Arial 10 pp is used for additional tips and notes about using the application

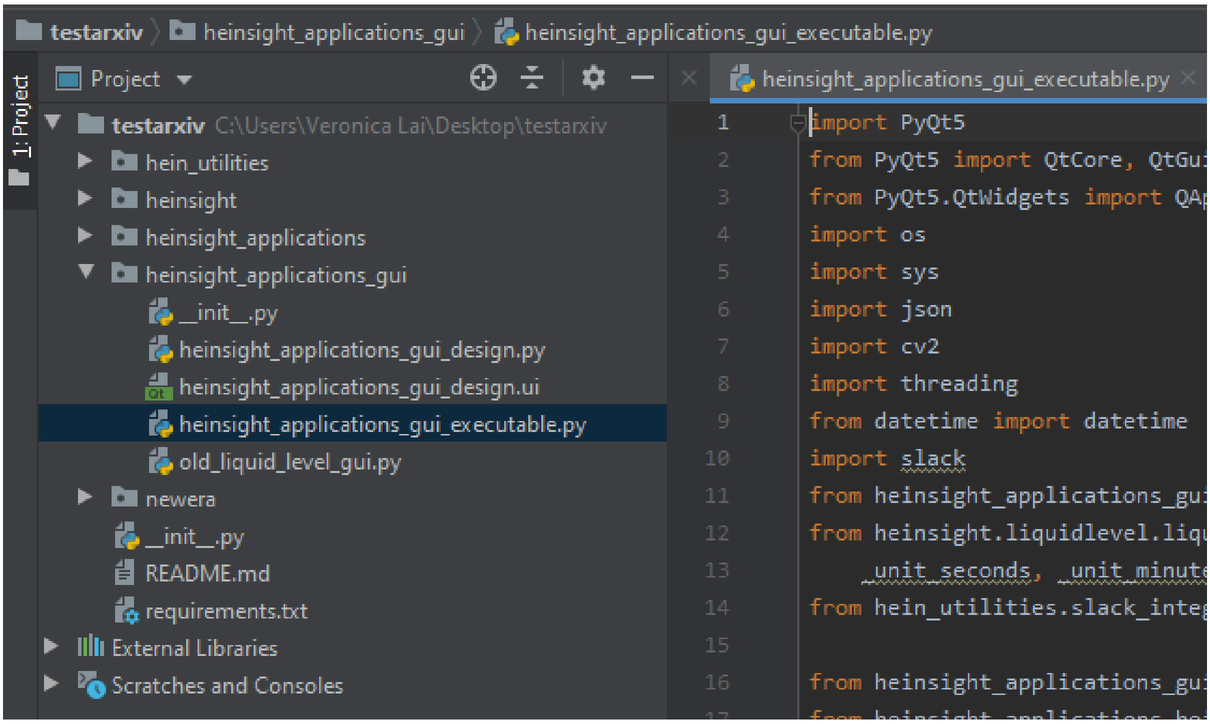
Setting up *HeinSight* on Your Computer

Open PyCharm and on the initial screen go to **Checkout/Get from Version Control > Git**, and use https://gitlab.com/heingroup/liquid_level_chemarxivi as the URL, and select the directory to save the project in. Alternatively if you are already in an open project, go to the top bar and select **VCS > Checkout/Get from Version Control > Git**. There are required Python packages that are required for running the scripts, and there should be a prompt to install these packages. Depending on personal preference, you may want to set up a project specific virtual environment; instructions on how to do so can be found here: https://www.jetbrains.com/help/pycharm-edu/creating-virtual-environment.html. Once the required packages have been installed, you will be able to run the graphical user interface to run one of the pre-designed automated experimental use cases. Included in the download is a copy of the required version packages that our group has developed to make everything work and includes: `hein_utilities`, `heinsight`, `heinsight_applications`, `heinsight_applications_gui`, and `newera`.

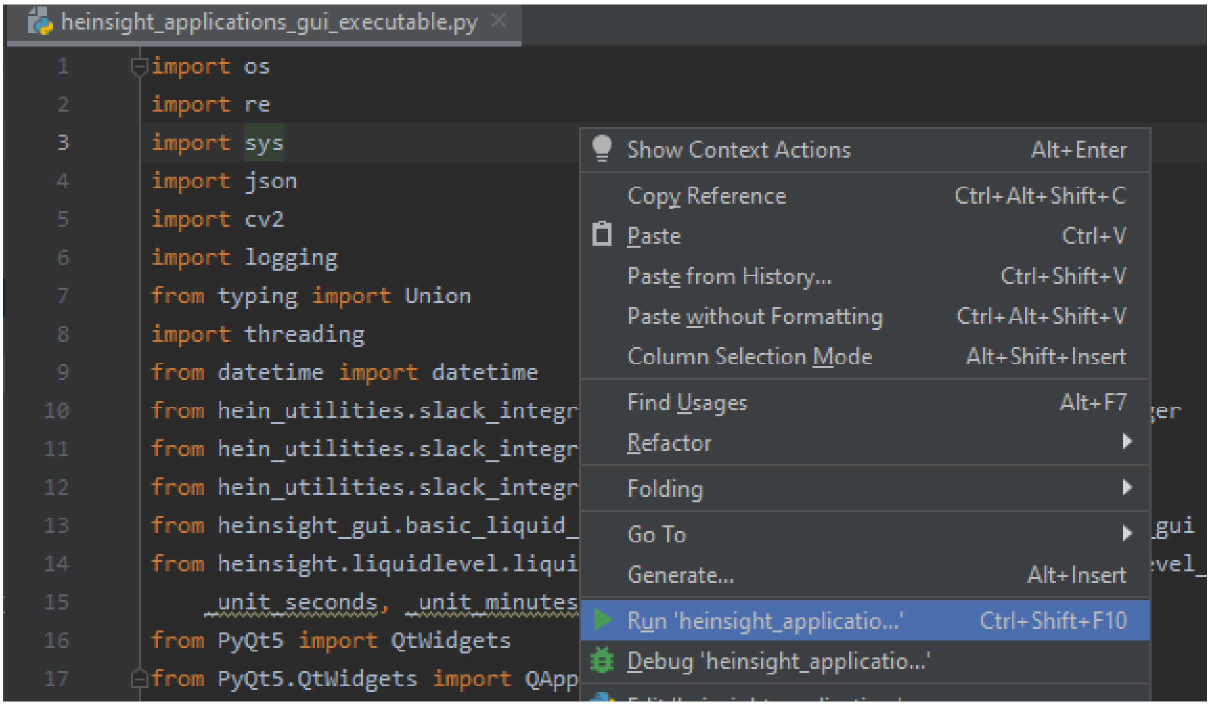
Opening/Running the GUI

1. Start PyCharm, and open the liquid level arxiv repository. If it is not yet open, go to **File > Open** to find the location, or **File > Open Recent**.

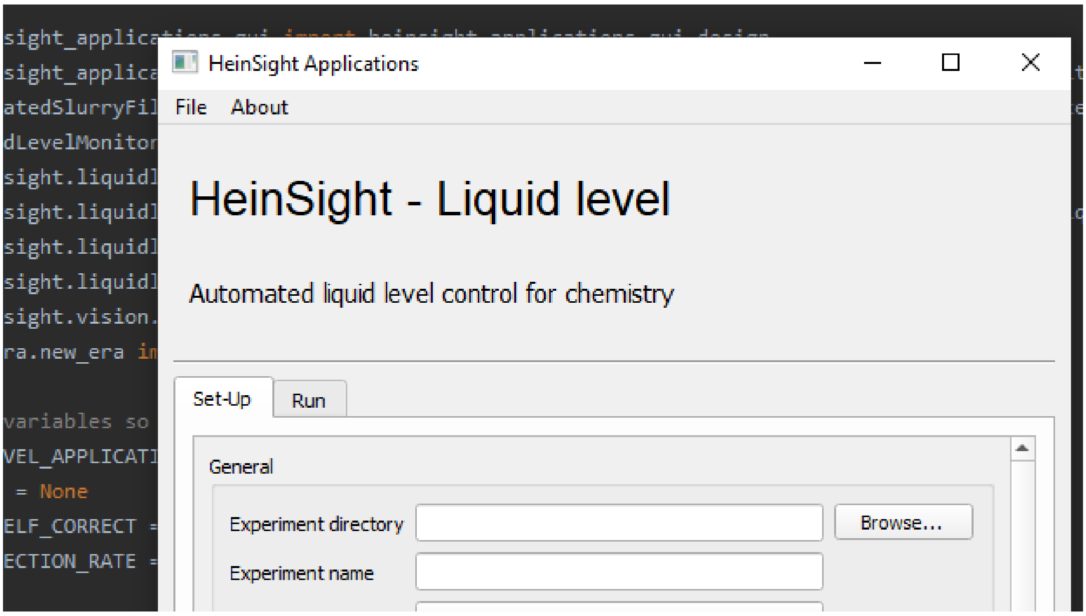
2. Navigate to the `heinsight_applications_gui\heinsight_applications_gui\heinsight_applications_gui_executable.py` file from the sidebar of PyCharm.



3. Right-click on the selected file and choose **Run** from the drop down menu.



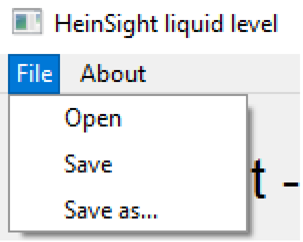
4. The GUI will open in a separate window.



You can fully customize, set-up, and run an experiment from this window. First, you will first need to set experimental parameters by filling out the “Set-up” tab. Once complete, you can move on to the “Run” tab. From here, you can initiate and run the experiment.

GUI “Set-up” Tab

Saving and loading a set-up file

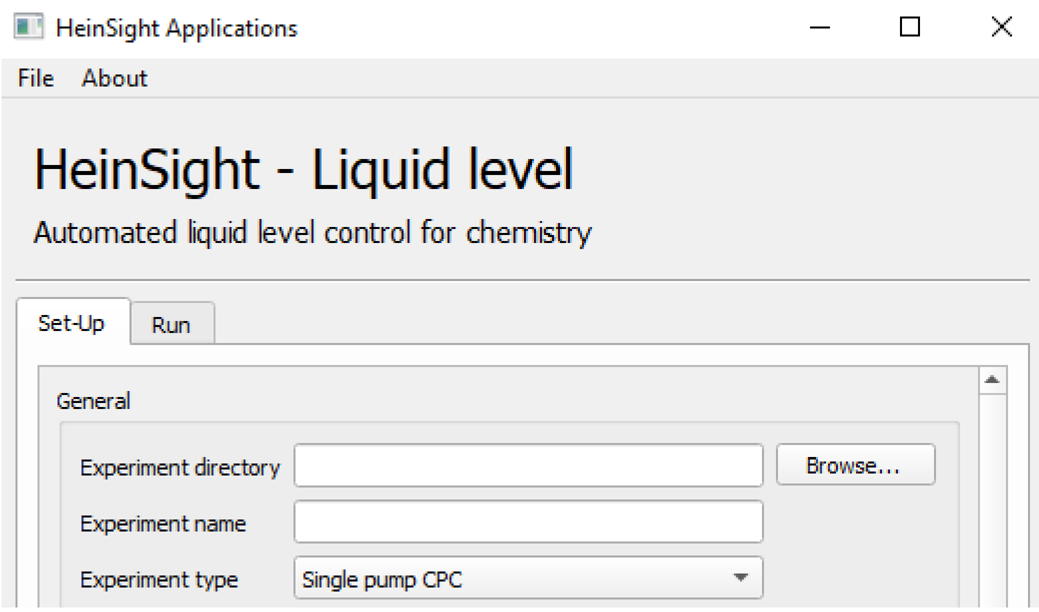


At any point, you can save the current set-up for the experiment as a JSON file so that it can be loaded easily at another time. Go to **File > Save as**, then select the location and name for the file.

If you loaded a file or saved a file, you can update it and update the save file by going to **File > Save**.

If you have a set-up file already created then you can go to **File > Open**, then select the set-up file to load into the GUI.

General Settings



Experiment directory

Click the browse button and navigate to the directory you would like all images taken by the experiment to be saved into. Within this experiment directory, a folder with the experiment name will be created, and images will be stored in sub-folders of this folder.

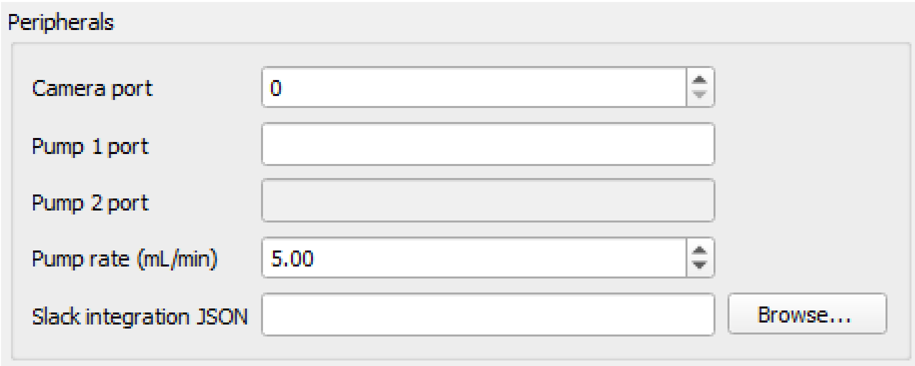
Experiment name

Type the desired experiment name (e.g. `CPC_EX_01`). This name is used to create a folder for subfolders of different types of images to be saved into throughout the experimental run.

Experiment type

Select the experiment type you want to run from the dropdown menu. Currently, the GUI supports “Single pump CPC,” “Dual pump CPC,” “Continuous distillation,” and “Filtration” applications.

Peripherals Settings



Camera port

Select the port number on your computer that the camera is connected to.

> Note: Generally, this is ‘0’ or ‘1’. The camera port is ‘0’ if there is no built-in camera on the computer and the webcam connected is the only webcam. The camera port is ‘1’ if there is a built-in camera and you are using an externally connected webcam to monitor the experiment. If multiple external webcams are connected, you will need to test and check which number corresponds with the camera monitoring the liquid level.

Pump port

To know which port a pump is connected to, **open** the Device Manager program from the Windows ‘Start’ menu, then look under ports. The ports are usually something along the lines of ‘COM#’ where # is a number.

Single-pump CPC experiment	Dual-pump CPC experiment
Only need to specify the port for the one pump in the experiment. The tubing needs to be set so that when the pump is set to dispense, liquid will be moved into the vessel being watched by the webcam.	Need to specify the ports for both pumps in the experiment. The tubing needs to be set so that when the pump is set to dispense, liquid will be moved into the vessel being watched by the webcam, for both of the pumps. The specification of the dispense and withdraw pump is only for readability, to know which pump will be set to dispense and which to withdraw throughout the entire

	experimental run.

Pump rate

Set the default pump rate (mL/min) to be used for the experiment. This is the rate that is used to transfer the liquid when a self-correction step does not have to be made to adjust the liquid level.

Slack Integration JSON

Click browse to choose a JSON file with the details to set-up a Slack bot.
This option allows you to receive updates and send simple commands (e.g. ‘Start’, ‘Pause’, etc) to control the experiment via a specific Slack channel. (See ‘Slack Control of an Experiment’ section of documentation for more information on commands.)

The JSON file should follow the following format:

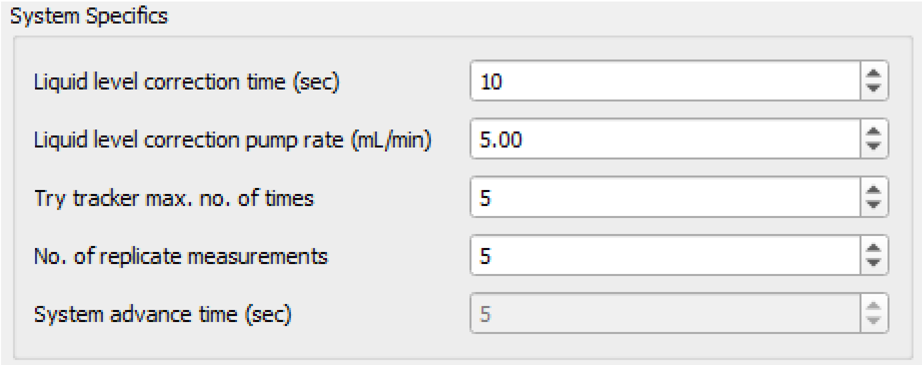
```
-------------- start of JSON file
        {
         "BOTNAME": "bot-name",
        "SLACKCHANNEL": "#slackchannel",
        "BOTTOKEN": "bot-token",
        "SLACKUSERTOKEN": "slack-user-token",
        "SLACKUSERNAME": "user-name",

        }
-------------- end of JSON file
```

where BOTNAME is the name of the bot, SLACKCHANNEL is the channel for the bot to send messages to, BOTTOKEN is the token for the bot, which must be found from Slack for the specific bot, SLACKUSERTOKEN is the token for the user that will interact with the bot, which must be found from Slack for the specific user as the member ID, and SLACKUSERNAME is the name of the Slack user. This file should not be shared online, as the BOTTOKEN should remain private to just the people in that Slack workspace.

For example:

```
-------------- start of JSON file
        {
         "BOTNAME": "gronkle",
        "SLACKCHANNEL": "#heinsight_experiments",
        "BOTTOKEN": "abcd-123456789123-123456789123-
dragonsdragonsdragons123",
        "SLACKUSERTOKEN": "ABCDEF123",
        "SLACKUSERNAME": "Hiccup",
        }
-------------- end of JSON file
```

System Specifics Settings



Liquid level correction time

Set the number of seconds the pump should run for when it needs to make a self-correction step.

Liquid level correction pump rate

Set the rate in mL/min the pump should be set at for when it needs to make a self-correction step.

It is recommended that this be set to a value similar to the default pump rate to reduce the possibility of cavitation occurring during the experimental run.

Try tracker max. no. of times

Set the maximum number of times that the application should try to find a liquid level before stopping the run should the algorithm fail to identify one.

> Note: The algorithm can struggle with finding a liquid level under certain conditions:
> - Lighting level changes or glare
> - The liquid level moves out of the selected region of interest (ROI)
> - The image taken to do the image analysis is not ideal for whatever reason
> - Low contrast between the liquid surface and the image background

No. of replicate measurements

Set the number of times the algorithm runs before determining the position of the liquid level.

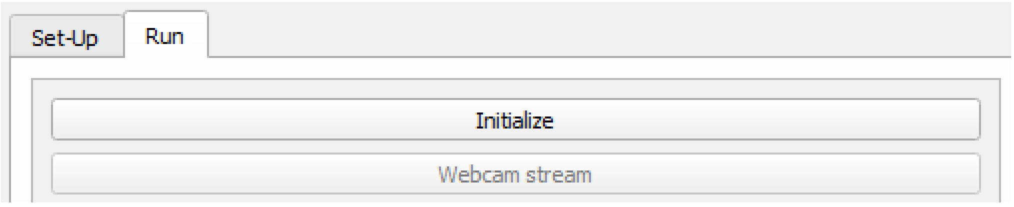
> Note: Positioning is based on the average of these images, discarding any outliers. This is to account for any small fluctuations or movements of the liquid in the vessel.

GUI “Run” Tab

The “Run” Tab requires that all hardware is initialized (connected to) before the experiment can be run via the ‘Start experiment’ button. .

If at any point you want to adjust the set-up of the experiment, click on the ‘Set-up’ tab and begin again. Then return to the ‘Run’ tab and go through the steps for the ‘Run’ tab from the beginning again.

Pre-steps



Initialize experiment

Click the ‘Initialize’ button. This will create the connection between the Python script and hardware (webcam, pump) using the parameters specified on the “Set-up” tab. This must be done regardless of experiment types.

> Note: If you go back to the “Set-up” tab, the first thing you do when you go back to the “Run” tab is press the initialize button again (even if you didn’t make any changes to the settings).

Any time after this initialize button has been pressed, you can open the ‘Webcam stream’ by **clicking** the ‘Webcam livestream’ button.

Using the Webcam Stream Window (*optional*)

Using the ‘Webcam stream’ button will allow you to select a region of interest (ROI) and determine how the liquid level detection algorithm is performing. This is optional but may be helpful to ensure the liquid level can be reliably found by the algorithm.

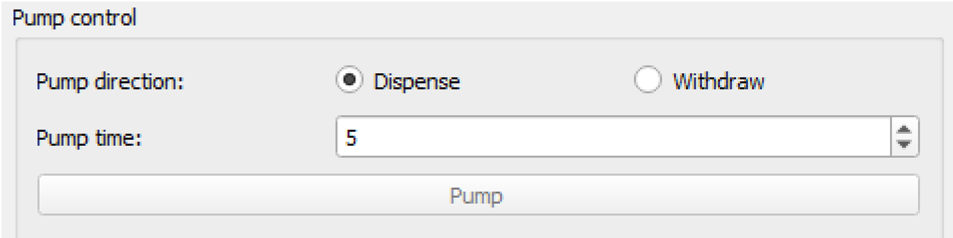
1. **Click** the ‘Webcam stream’ button. This will open a new window that allows you to see the raw footage as well as the effects that the liquid level finding algorithm is having in real-time.

2. **Click** the ‘select region of interest’ button (in the new window). A still image pop-up. From here, you can select the ROI in which you would like to algorithm to look.

3. Use the mouse to **draw** any closed polygonal shape by clicking where you want each corner to be located. It is best to keep the shape as simple as possible. When you are satisfied with the selection, **press ‘c’** then the **‘enter’ key**. If you are unsatisfied or make a mistake, **press ‘r’** and the selection process will reset.

   Note: When finished, you will be returned to the main window and see the selected ROI drawn atop the webcam footage. You can toggle the outline drawn on the webcam view by clicking the 'show mask contour on image' button. If you only want to see what is within the selected region of interest, toggle the 'show area in mask' button. To see how the liquid level algorithm is performing, toggle the 'show liquid level lines' button.

Pump Control



You can trigger the pump from the GUI to adjust the liquid level so that it is better positioned for your experiment. This can be helpful if you need to place the webcam at a specific height or are trying to avoid a possible false trigger for the algorithm (e.g. stir bar, strong light glare, etc.)

Select whether you would like the pump to 'Dispense' or 'Withdraw' and **set** the number of seconds you would like it to run. Then **press** the 'Pump' button.

> Note: Direction should be set with respect to the vial being watched. If you are running a 'Dual-pump CPC' experiment, the command will be sent to Pump 1.

Run an Experiment

Start an Experiment

Click the 'Start' button to start an experiment. This will prompt additional pop-up windows that will allow you to set the ROI and tolerance zones (see below).

> Note: At this point the GUI window will close and separate interactive windows will open.

Selecting an ROI

The first window to appear allows you to select a region of interest (ROI) in which the liquid level finding algorithm will search.

1. Use the mouse to **draw** any closed polygonal shape by **clicking** where you want each corner to be located. It is best to keep the shape as simple as possible.
2. When you are satisfied with the selection, **press 'c'** then the **'enter' key**. If you are unsatisfied or make a mistake, **press 'r'** and the selection process will reset.

Setting a Reference

The reference indicates the ideal height of the liquid level. Note that this is only required for the CPC type experiments and not for filtration or distillation.

1. Once the ROI has been selected, a new raw image will appear. **Set** the reference level by **clicking** on the image at the desired height.

2. When you are satisfied with the selection, **press 'c'** then the **'enter' key**. If you are unsatisfied or make a mistake, **press 'r'** and the selection process will reset.

Note: It is best to set the reference at the vertical middle of the ROI. This gives the liquid level finding algorithm the best chance of correctly locating the meniscus.

Setting Tolerance and Fail-Safe Zones

Tolerance zones determine at what point any corrective action should be taken. There are two levels. The first dictates the zone outside of which self-correction should happen. The second is a fail-safe measure and is only applicable to Dual-pump CPC. Pumping will be suspended if the liquid level is determined to be outside the fail-safe zone.

1. **Set** the tolerance zone by **clicking** on the image where you would like to upper and lower bounds to be.

2. When you are satisfied with the selection, **press 'c'** then the **'enter' key**. If you are unsatisfied or make a mistake, **press 'r'** and the selection process will reset.

3. Repeat steps 1 and 2 to set where the fail-safe bounds should be located.

The experiment will begin after all necessary selections have been made.

Monitoring an Experiment while Running

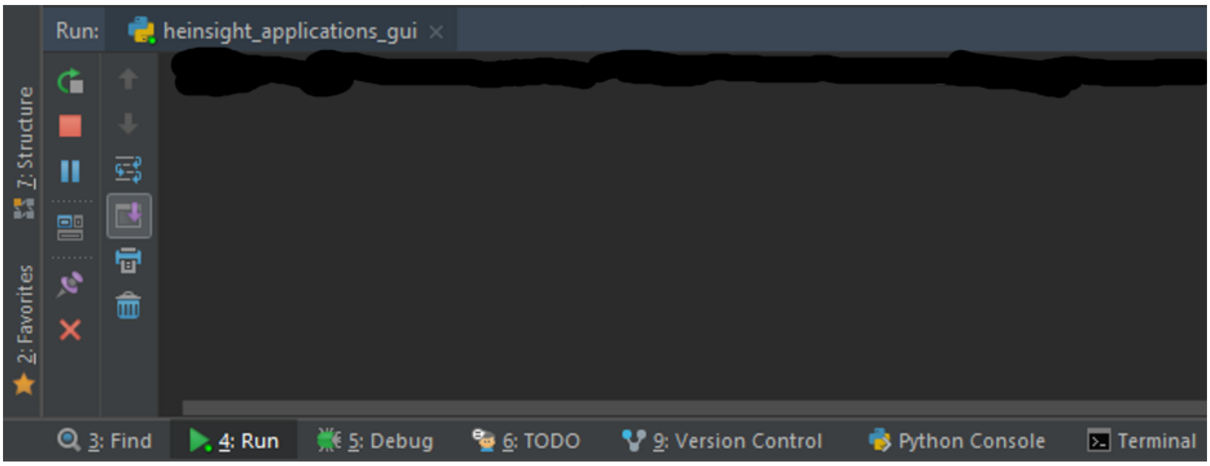
There are three ways to monitor an experiment while it is running: 1) Slack, 2) PyCharm, 3) File explorer.

Monitoring through Slack

If you have set up a Slack bot, you will be able to see updates in the specified Slack channel and control the experiment by sending messages to the Slack channel that was specified in the setup file; see Slack commands for instructions on how to send commands via Slack messages.

Monitoring through PyCharm

You can also monitor the experiment by looking at the statements printed to the console in PyCharm. **Click** the 'Run' tab located in the bottom left corner of the PyCharm window.



Monitoring through the file explorer

To see all images generated during an experiment, **navigate** to the 'experiment directory' you set and **open the folder** with the current 'experiment name.' You will see 3 different folders of images:

`raw_images`: contains all raw images captured by the webcam

`all_drawn_images`: contains images with lines drawn by the level-finding algorithm

`slack_images`: contains all images sent to Slack

Observing the `all_drawn_images` folder is a good way to see how the liquid-level is changing and how well the algorithm is performing without moving the webcam.

> Note: Do not leave the folder where all the images are being saved open while the experiment is running if you are not actively viewing it. Doing so may overload the computer memory, which will cause the application to quit.

End an Experiment

All experiments types are set by default to run indefinitely. The only reasons why an experiment might stop are:

1) there is an error due to hardware failure or loss of internet connection (if Slack integration was enabled), or
2) a liquid level cannot be found after the set max number of consecutive attempts.

End an Experiment through Slack

The best way to actively end an experiment is through Slack. **Use** the 'end experiment' command, which will end the script and stop the pump from running.

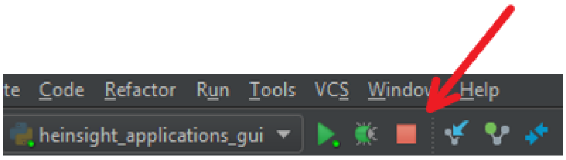
End an Experiment Manually

An experiment can also be ended manually.

1. First, **press** the 'Start/Stop' button on the pump(s). If you are running a Dual-Pump CPC experiment, you must stop both.

    > Note: The pump will make a beeping sound after the 'Start/Stop' has been pressed. Simply press any other button to turn this off.

2. Then, **press** the red square button in the 'Run' tab of PyCharm or **click** the red square button at the top of the program in the toolbar. This will stop the script.



Details on how to run the different experiment types

The different experiments require different steps to be performed. Regardless of the experiment, there are many basic steps that are required to be performed by the different experiments; consult the Basic steps to run an experiment section below for details, and be sure to look through the Tips and recommendations section.

Basic steps to run an experiment

Some of the steps for the different experiment types and configurations may be identical, and they are included here for easy reference

Select a region of interest

Click points on either window to draw any polygonal shape to set as a region of interest for the liquid level algorithm to work within. You do not need to connect the first and last points together, this will happen automatically. When you are satisfied with your selection, press 'c', then press the 'enter' key, which will show you your selection, then press the ‘enter’ key again. If at any point while selecting the region of interest you are not satisfied, to reset all the current selections press 'r'.

Select the reference level

For the CPC experiments, this reference is required so the system knows what direction to pump to in order to adjust the liquid level to move within the tolerance zone.The reference level needs to be between the two levels. For the filtration and distillation type of experiments this step is not required.

Select a single point on the image, and this will cause a line to be drawn on the image that will be set as the reference line. When you are satisfied with your selection, press 'c', then press the 'enter' key, which will show you your selection, then press the ‘enter’ key again. If at any point while selecting the region of interest you are not satisfied, to reset all the current selections press 'r'.

Select the tolerance bound/bounds

Depending on the experiment type, you may need to select one or two tolerance bound/bounds.

If you need to select a single tolerance bound, what this means is you need to select a single lower level limit, so that when the liquid level goes underneath your selected level the self-correction step will be made to adjust the liquid level to a higher level; this is the case for filtration and distillation.

If you need to select two tolerance bounds, this means you need to select an upper and lower bound that will contain the allowed heights that the liquid level can be at. So if the liquid level travels outside of these two lines, either above the upper line or below the lower line, then the self-correction step will be triggered. This is the case for the CPC experiments.

To select a tolerance level, click one of the images. This will cause a line to be drawn on the image that will be set as the tolerance bound line. When you are satisfied with your selection of either the single tolerance bound or both bounds, press 'c', then press the 'enter' key, which will show you your selection, then press the ‘enter’ key again If at any point while tolerance bound/bounds, to reset all the current selections press 'r'.

Select the fail-safe tolerance bounds

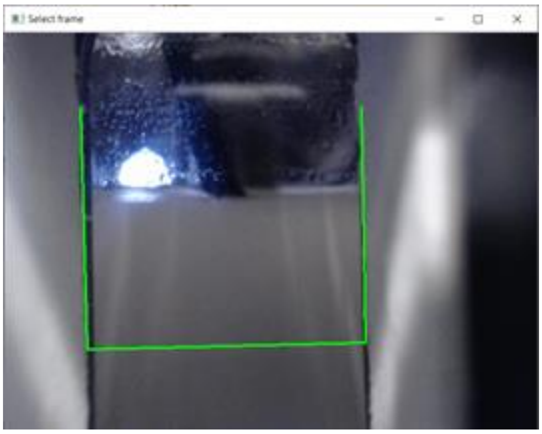
The fail-safe tolerance bounds are two bounds, where if the liquid level goes above the upper bound or below the lower bound, then the program will stop and the experimental run will end.

For now it is recommended to set these outside of the region of interest, as the region of interest is generally tight enough that if the liquid level were to move outside of the region of interest, that the program will end because of that then.

To make a selection, click on either of the images. When you are satisfied with your selection of both bounds, press 'c', then press the 'enter' key, which will show you your selection, then press the ‘enter’ key again If at any point while selecting the bounds you are not satisfied, to reset all the current selections press 'r'.

Tips and recommendations

For selecting the region of interest, it is recommended that if the vessel is roughly rectangular shaped, that the region of interest also is a roughly rectangular shape made up with the least number of points as possible; in this case the optimal region of interest would be made of four points. Below is an example of a well selected region of interest.



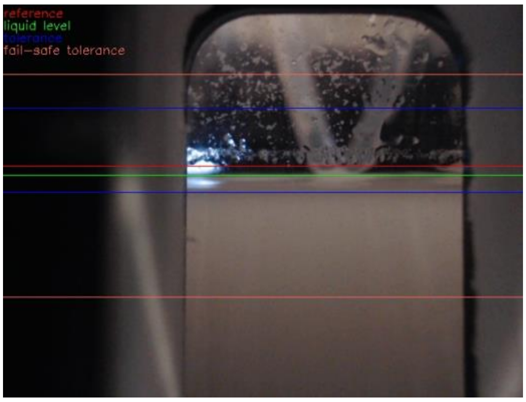
Depending on the lighting of the set up, the region where the liquid level is likely to hover/drift within is recommended to be in the middle of the image, as this gives the equal space both above and below for liquid level drift. This way, if the algorithm incorrectly identifies the location of the liquid level, there is leeway on both sides for some drift to occur before the system can correct for the drift. Similarly, it is recommended that the reference level is set to a level roughly in the middle of the image.

The choice of distance away from the reference level that the tolerance level(s) is/are set is recommended to not be too far away. They need to be within the region of interest, but should not be too close to the edges of the region of interest, so if there are instances when the liquid level algorithm mistakes the location of the liquid level, that it will not move out of the region of interest the next time the webcam is used to monitor the liquid level height.

Having tolerance levels closer to the reference level means a higher frequency of corrections to the liquid level will be made, and distance that the liquid level can drift to is smaller. You must also ensure during set up that the tolerance levels will allow for the self correction step to occur without the liquid level from drifting too far in the other direction.

Overall it is recommended that the tolerance levels are selected closer, rather than farther, away from each other and the reference line.

This is an example of choosing the reference and tolerance levels to be in the most optimal position for a dual pump CPC system:



In this example the reference level is selected to be slightly above the middle point of the image, as it was seen that partially covering the light at the back of the instrument produced better images for the algorithm, and because the experiment required a high level of volume overall split between two vessels. The lower tolerance level because during past runs with this set up we noticed that the liquid level would generally drift downwards. Both tolerance levels have been selected so that after a self correction step has occurred, the liquid level will move to be within the tolerance zone.

Single pump CPC

Run CPC using 1 peristaltic pump; the purpose of the system is to maintain the liquid level in a vessel between two user set levels. The tubing set up for the pump needs to be arranged so that when the pump is set to dispense, liquid will be dispensed into the vessel that is being watched by the webcam. The user will need to select a region of interest for the liquid level algorithm to work within, and specify a reference liquid level and the two tolerance levels that the liquid level will not be allowed to move outside of. Then when the experiment is started, the pumps will cycle to transfer liquid between the two vessels. Ideally there will be no liquid level drift, but if there is drift of the liquid level above the tolerance level, then the pump will be stopped, self-correct is triggered to withdraw from the vessel being watched to move the liquid level downwards back into the tolerance zone. After the self-correction step, liquid cycling resumes. Similarly, if the liquid level goes below the lower tolerance level, the self-correction step will dispense liquid into the vessel being watched. This continues indefinitely, or until the program is manually ended, to maintain the liquid level between the two tolerance levels.

	Step
1	Press the ‘Start experiment button’
2	Select the region of interest
3	Select the reference level
4	Select two tolerance bounds
5	Start

Dual pump CPC

The way the dual peristaltic pump cpc system works is that it is used to run CPC using 2 peristaltic pumps; the purpose of the system is to maintain the liquid level in a vessel between two user set levels. With both pumps set up, the tubing of both pumps need to be arranged so that when they are both set to dispense, liquid will be dispensed into the vessel that is being watched by the webcam. The user will need to select a region of interest for the liquid level algorithm to work within, and specify a reference liquid level and the two tolerance levels that the liquid level will not be allowed to move outside of. Then when the experiment is started, both pumps will run, with one dispensing liquid into the vessel being watched and the other withdrawing liquid from the vessel being watched. Ideally there will be no liquid level drift, but if there is drift of the liquid level above the tolerance level, then both of the pumps will be stopped, and the pump that is used to withdraw liquid out of the vessel being watched will be triggered to self correct for the too high liquid level. After the self correction step, both pumps are started again. Similarly, if the liquid level goes below the lower tolerance level, then the pump that is set to dispense into the vessel being watched will be used for self-correction. This continues indefinitely, or until the program is manually ended, to maintain the liquid level between the two tolerance levels.

	Step
1	Press the ‘Start experiment button’
2	Select the region of interest
3	Select the reference level
4	Select two tolerance bounds
5	Select two points on the image for the fail-safe tolerance
6	Start

Continuous distillation

Run a continuous distillation experiment using a single peristaltic pump. The purpose of the system is to maintain the liquid level in a vessel above a single set tolerance level. The tubing set up is so that when the pump is triggered to dispense, liquid gets added into the vessel. The user will need to select a region of interest for the liquid level algorithm to work within, and specify a single tolerance level that the liquid level will not be allowed to move below. Then when the code is started, heating will either have already been going, or will have to be started. The system will continuously monitor the liquid level until it falls beneath the tolerance level. Once this happens, self correction is triggered to dispense liquid into the vessel to move the liquid level above the tolerance level. This continues indefinitely, or until the program is manually ended, to maintain the liquid level above the tolerance level.

	Step
1	Press the ‘Start experiment button’
2	Select the region of interest
3	Select one tolerance level
4	Start

Filtration

Run a filtration experiment using a single peristaltic pump. The purpose of the system is to maintain the liquid level in a funnel above a single set tolerance level. The tubing set up is so that when the pump is triggered to dispense, liquid gets added into the funnel. The user will need to select a region of interest for the liquid level algorithm to work within, and specify a single tolerance level that the liquid level will not be allowed to move below. Then when the code is started, the system will continuously monitor the liquid level until it falls beneath the tolerance level. Once this happens, self correction is triggered to dispense liquid into the funnel to move the liquid level above the tolerance level. This continues indefinitely, or until the program is manually ended, to maintain the liquid level above the tolerance level.

	Step
1	Press the ‘Start experiment button’
2	Select the region of interest
3	Select one tolerance level
4	Start

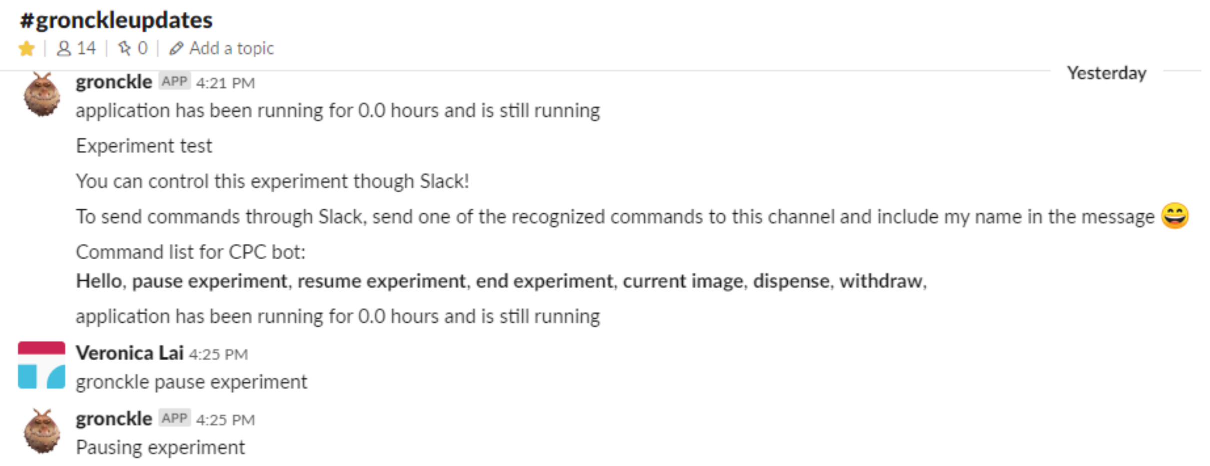
Slack control of an experiment

If an experiment has been set up with a Slack bot, it is possible to control the experiment through Slack. On the channel where the updates are sent, send ‘help’ to the channel to get help on the commands to control the experiment.

Slack commands

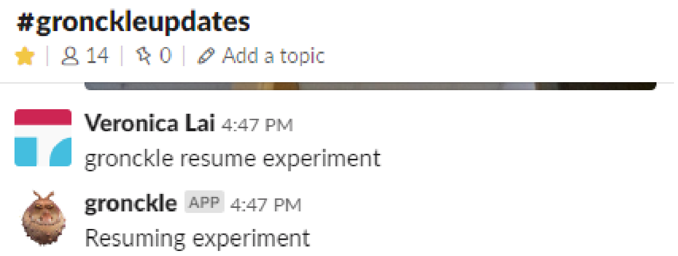
pause

Pause a running experiment and the pump(s)



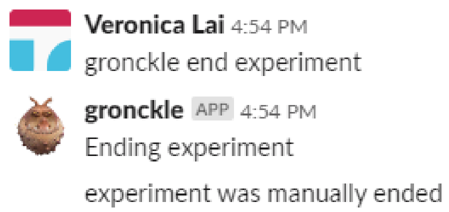
resume

Resume a paused experiment and the pump(s)



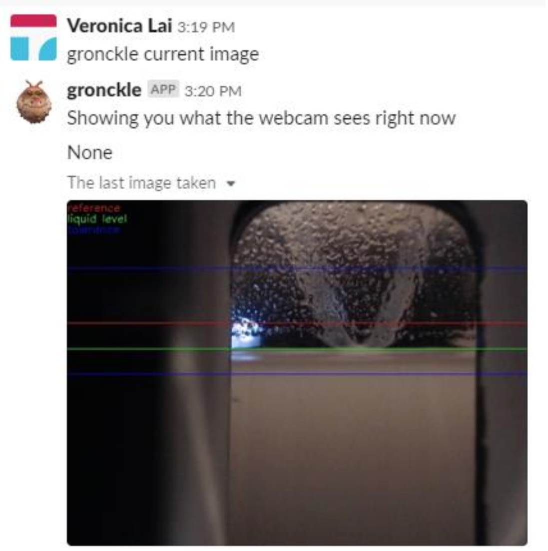
stop

End an experiment and stop the pump(s)



current image

Get and post the current image from the webcam to Slack



dispense [n]

Uses either the only pump, or the pump that is used for dispensing if there are two pumps setup, to dispense into the vessel being watched at the self-correction pump rate for n seconds specified in the command message.

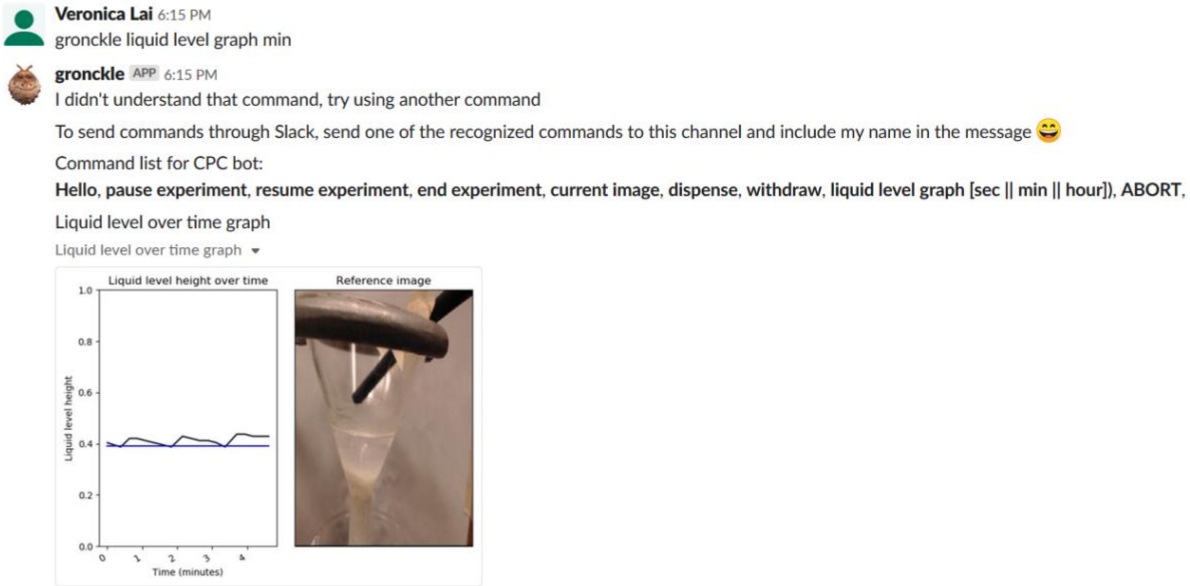
withdraw [n]

Uses either the only pump, or the pump that is used for withdrawing if there are two pumps setup, to withdraw from the vessel being watched at the self-correction pump rate for n seconds specified in the command message.

liquid level graph [sec | min | hour]

Make and post a graph of the liquid level over time for the current experiment. Specify the units of the x-axis with one of the keywords in brackets.

The graph will be plotted side-by-side to an image of the setup (the first image that was taken) so that the y axes are aligned. The y-axis is the relative liquid level height (relative to the height of the image), where the y value in the graph corresponds to the height that the liquid level was detected in the image.



The resulting graph: