# Combining automated microfluidic experimentation with machine learning for efficient polymerization design

**README file**

Benjamin A. Rizkina, Albert S. Shkolnika, Neil J. Ferraroa and Ryan L. Hartmana,\*

**1. System requirements:**

Dependencies and operating systems:

1. Windows 10\*
2. National Instruments LabVIEW ® 2019
3. Mathworks MATLAB® R2019a
4. LINX by Digilent/LabVIEW MakerHub
5. ICI® camera drivers
6. Elvesys® pump drivers
7. Jan Bischof’s MATLAB-LabVIEW TCP-IP data exchange toolkit
8. Minimum quad core processor and 16 GB RAM
9. Monitor running at 3360x2100 resolution (optional but recommended)

Versions the software has been tested on:

\*This software has been tested on Windows 10. The analysis code has also been successfully implemented and tested on MacOS Catalina 10.15.

Required non-standard hardware:

To perform the experiments additional hardware is required. A full outline is provided in the Supplemental Information. This includes but is not limited to:

* Elvesys® OB1 pressure controller
* Arduino Mega microcontroller
* Relay boards
* Stable 12V and 5V power supplies
* Peltier cells and heat removal system
* Pulsed dosing pumps
* Manifolds
* Thermal camera
* Inert environment glove box
* Chemical handing and ventilation equipment

**Included files and descriptions:**

|  |  |  |
| --- | --- | --- |
| **File name** | **Description** | **Proprietary license** |
| Main\_Code.m | Main data analysis code |  |
| Conversionscript.m | Code for generating the conversion from the experimental exotherm |  |
| RateConstants.m | Code for computing the rate constants |  |
| MATLAB\_workspace.m | Workspace with all data featured in publication |  |
| Results from all trials.xlsx | Database of raw data from all experiments |  |
| Final experimental workup.xlsx | Worksheet for reagent preparation |  |
| Arduino\_analog\_read.vi | Read analog data from Arduino | ✔️ |
| Arduino\_analog\_write.vi | Write analog data to Arduino | ✔️ |
| Arduino\_logic\_array.vi | Arduino array conversion | ✔️ |
| CameraNUC.m | Infrared camera nonuniformity correction | ✔️ |
| Control\_project.aliases | LabVIEW® project files |  |
| Control\_project.lvlps |  |
| Control\_project.lvproj |  |
| exchangeData.m | Exchange data between MATLAB® and LabVIEW®, MATLAB® side | ✔️ |
| exchangeData.vi | Exchange data between MATLAB® and LabVIEW®, LabVIEW® side | ✔️ |
| FrameGrabber.h | ICI® library for grabbing frames from camera | ✔️ |
| ici9000Matlab.m | Example MATLAB® code for IR camera | ✔️ |
| license.txt | License agreement |  |
| main\_IR.m | Main MATLAB® file for running IR camera | ✔️ |
| MATLAB\_comm.vi | Example communication file between MATLAB® and LabVIEW®, | ✔️ |
| Str2Exp.vi | Convert a string of text to experimental instructions |  |
| uAIR control v7.vi | Main user interface |  |

**2. Instillation guides:**

To perform experiments:

1. Build experimental system as outlined in the SI
2. Install LabVIEW® 2019
3. Install MATLAB® R2019a
4. Install all requisite proprietary drivers provided with thermal camera and pumps
5. Install LINX for LabVIEW®
6. Flash firmware to Arduino Mega
7. Check all wiring connections
8. Use either native ICI® software or MATLAB® file “main\_IR.m” for thermal camera measurements
9. Launch “uAIR control v7.vi” and check for any compilation errors
10. Begin experiments

Typical installation and configuration time: about 4 hours

To only analyze data:

1. Insure MATLAB R2019a is installed and properly configured

Typical installation and configuration time: between 0-30 minutes

**3&4. Demo and instructions for use:**

To reproduce presented results from preprocessed data:

1. Place “Main\_Code.m” and “MATLAB\_workspace.m” in the same folder
2. Open “Main\_Code.m”
3. Figure 5**a**: Run lines 310-330
4. Figure 5**b**: Run lines 185-214 using option 2
5. Figure 5**c** and **d**: Run lines 331-353
6. Figure 6 top: Run lines 238-272
7. Figure 6 bottom: Run lines 299-309

**Expected output**: matches figures and data presented in the publication

**Expected run time:** around 5 minutes

To reanalyze data:

1. Open “Main\_Code.m” file
2. Place “Results from all trials.xlsx”, “conversionscript.m” and “RateConstants.m” in the same working folder
3. To generate a sample experimental plan (randomized) run lines 1-14
4. Initialize workspace by running lines 15-21
5. Import data and set constants by running lines 22-88
6. Run “conversionscript.m” and “RateConstants.m”
7. Train different neural networks by running lines 89-136
8. Plot the error percentage heatmap by running lines 137-155
9. Experiment with different network types for training and test loss by running lines 156-166
10. Make network performance plot by running lines 167-175
11. Optionally view individual error plots by running lines 176-187
12. Select which network architecture you would like to use and change lines 193 and 194 accordingly
13. Choose the desired accuracy level and change line 198 accordingly, higher accuracy will take longer to train
14. Run lines 188-207 to train the final network
15. View error over test indices by running lines 208-211
16. Compare AI and experimental data by running lines 212-241
17. Perform AI experimental space mapping by running lines 242-264
18. Plot AI results by running lines 265-299
19. Generate temperature range dataset by running lines 300-325
20. Make temperature range plot by running lines 326-336
21. Plot the experimental sequence by running lines 337-357
22. Plot the kinetic rate constants by running lines 358-380
23. Optionally plot the concentration versus time plots by running lines 381-392
24. Find optimal reaction conditions by running lines 393-418
    1. To find profits enter accurate prices in lines 397-401
25. To perform Digital Twin optimization as in the SI run lines 419-618. Note that the gradient algorithm is prone to false maxima and care should be taken when adapting this code.
26. Plot Digital Twin results by running lines 619-642
27. Optionally create an animation by running lines 643-712

**Expected output**: matches figures and data presented in the publication, might have slight deviations in AI generated results due to randomness in training

**Expected run time:** around 24 hours