User's Manual for the Stanford Microfluidics Lab CDI Program

Tristan D. Hasseler
June 2020

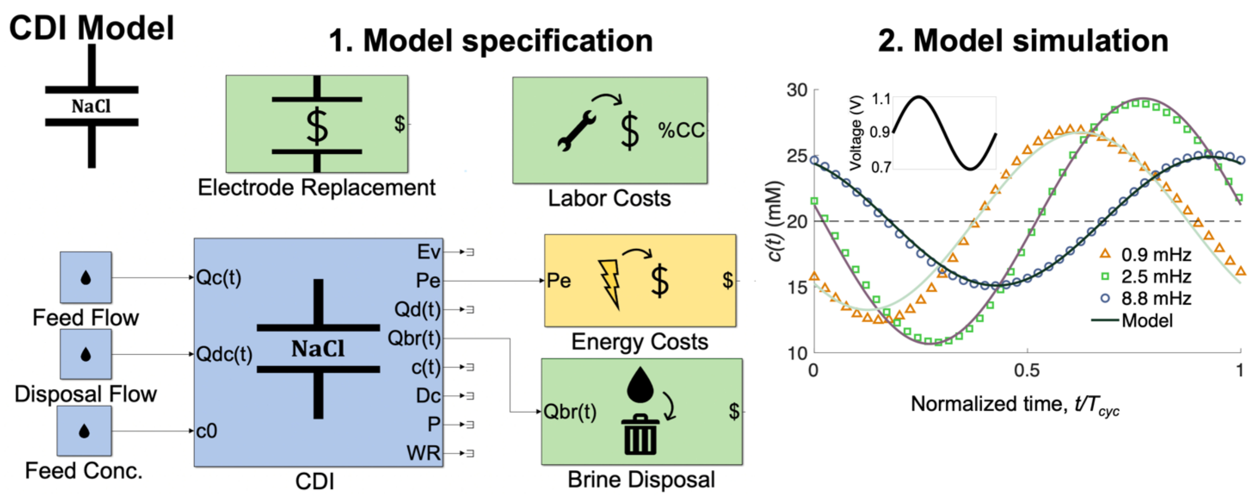

Table of Contents

1. User instructions for running the CDI program 3
   1.1. Downloading and running the CDI program 3
   1.2. Specifying configurations in Simulink 3
   1.3. Specifying plant parameters and operating conditions in the GUI 6
   1.4. Running standalone simulations 8
   1.5. Running economic studies 8
   1.6. Running in batch mode 10
   1.7. Running in constant current (CC) mode 11
2. Glossary of workspace variables 12
   2.1. ModelParams 12
   2.2. SimulationOutputs 13
   2.3. SimulinkInputs 15
   2.4. EconomicOutputs 16
   2.5. Batch Mode Outputs 16
3. References 18

1. User instructions for running the CDI program

We here describe the process of running the CDI program and basic simulations. Sections 1.1-1.2 respectively outline how to download the required files and configure the Simulink workspace. Section 1.3 details how plant parameters and operating conditions are specified in the GUI program. Lastly, Sections 1.4, 1.5, and 1.6 respectively show how to run standalone, economic, and parameter batch mode studies with the program.

The material presented here is developed in full in Hasseler et al. (2020) in the references section. Please consult this for relevant definitions, theory, and discussion.

1.1. Downloading and running the CDI program

The CDI Program is provided as a free, downloadable Matlab program. Please visit http://microfluidics.stanford.edu/download/index.html to download the installer and example files.

Requirements:
- Matlab version **R2019b** or later.
- Simulink must be installed to run the CDI Program.

Installation instructions:
1. Download the Matlab app installer file *CDI_Program.mlappinstall.*
2. Move the file to a desired folder in your Matlab path.
3. In Matlab, navigate to the current folder where the file is stored.
4. In the "*current folder*" navigation pane in Matlab, double-click on the app installer file.
5. Click on "*Install*" on the pop-up prompt.
6. Navigate to the "*Apps*" tab in Matlab and click on "*CDI_Program.*"
7. In the program, navigate to the "*Sim Parameters*" tab and specify the name of the Simulink file you would like to analyze. The Simulink file must be in the same active folder. An example Simulink file *Example_CDI_Model.slx* has been provided in the download file.

1.2. Specifying configurations in Simulink

As the first step, the user must specify a desired system configuration in Simulink. This allows the CDI program to load the appropriate economics subsystems for analysis and fully define the CDI system.

1. Open the Simulink program titled *Block_Library.slx.* Here you will see all of the currently developed subsystem blocks that are available to build the CDI plant. Table 1 lists available water, energy, and economic subsystem blocks provided in the CDI block library.

Table 1. Available subsystem blocks

Type	**Block**
Water systems	CDI cell
	Water flow
	Feed concentration
Energy systems	Grid energy source*
	Battery energy source*
	Solar panel energy source*
Economics systems	Electrode replacement*
	Waste disposal*
	Labor/Maintenance*

*requires a to workspace block connected to output (see Table 3)

2. Create a new blank Simulink model.
3. Navigate to *Modeling* → *Model Settings* → *Solver*. Change the variable titled "*Start time*" to "*SimulinkInputs.t0*" and *"Stop time"* to *"SimulinkInputs.tfinal"*. Next, under *"Solver Selection"* change "*Type*" to *"Fixed-step."* Lastly, under "*Solver details"* change the variable titled *"Fixed-step size (fundamental sample time)"* to *"1.0"*. Click apply to close the settings window.
4. Copy and paste the CDI cell block from *Block_Library.slx* into the new blank model. Table 2 describes the inputs and outputs associated with the CDI cell block.

Table 2. Inputs and outputs for the CDI cell Simulink block

Model inputs	**Model outputs**
*Qc(t)** - Charging flowrate [m^3 s^{-1}]	*Ev* - Volumetric energy consumption [kWh m^{-3}]
*Qdc(t)** - Discharging flowrate [m^3 s^{-1}]	*Pe* - Power requirement [kW]
*c0** - Feed concentration [mM]	*Qd(t)* - Desalinated water flow [m^3 s^{-1}]
	Qbr(t) - Brine/wastewater flow [m^3 s^{-1}]
	c(t) - Effluent concentration [mM]
	Dc - Desalination depth [mM]
	P - Productivity [L m^{-2} h^{-1},]
	WR - Water recovery ratio

*connection is required to run program

5. Copy and paste other desired blocks from *Block_Library.slx* into the new model.

6. Connect desired subsystems. Note: as a minimum to run the CDI program, a CDI cell block must be included. Additionally, water flow blocks must be fed into the CDI model through the *Qc* and *Qdc* input ports. A feed concentration block must also be fed into the *c0* port. Fig. 1 shows a sample CDI system configuration that includes all necessary input blocks for the CDI cell block plus electrode replacement, energy costs, and brine disposal blocks.
7. Double click on each periphery block and specify any required user inputs within, indicated as blocks within blue panels. **Do not alter any blocks that are not specified as a user input.**

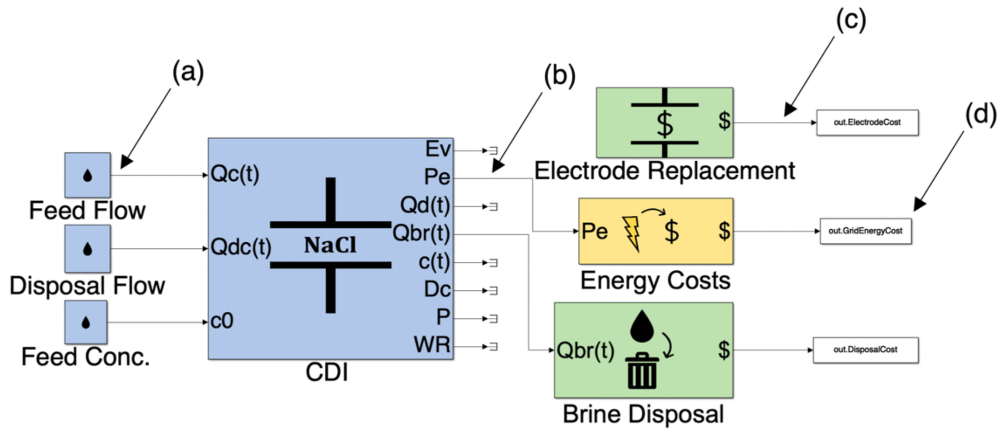

Fig. 1. Example CDI system architecture designed in Simulink with energy cost, waste disposal, and electrode replacement subsystems. (a) Water flow stream [$m^3\ s^{-1}$], (b) power stream [kW], (c) capital stream [$], (d) to workspace block, which allows Simulink to send data to the Matlab workspace, where data is then read into the program and presented in the GUI.

8. Outputs from blocks other than the base required blocks discussed in step 6 require a "to workspace" block in order to send data to the CDI program GUI. See Table 3 to determine which blocks require a "to workspace" block. Callout (d) in Fig. 1 shows an appropriate to workspace block connection.
9. Connect a to workspace block to each block that is needed. Double-click on the to workspace block. Change the selection for *"Save format"* to *"Timeseries."* Depending on which subsystem the to workspace block is connected to, the appropriate *"Variable name"* must be typed. Table 3 lists the required variable names for each subsystem block.
10. Save the Simulink model with a desired filename.

Table 3. Required to workspace variable names

Subsystem block	**Required to workspace variable name***
Electrode replacement	"ElectrodeCost"
Grid energy source	"GridEnergyCost"
Battery energy source	"BatteryEnergyCost"
Solar panel energy source	"SolarEnergyCost"

Waste disposal	"DisposalCost"
Labor costs	"LaborCost"

*Case and space sensitive

1.3. Specifying plant parameters and operating conditions in the GUI

Once the desired configuration has been specified in Simulink, the CDI plant parameters and operating schemes can be specified in the graphical user interface (GUI).

1. In the Matlab apps tab, open the app titled *"CDI Program."*
2. Once the program has launched, in the "*Sim parameters*" tab, specify the name (including the .slx extension) of the Simulink model file created previously. Ensure the Simulink file is in the present working directory.
3. Table 4, Table 5, and Table 6 detail baseline parameters that must be entered by the user to run the base program.

Table 4. CDI plant inputs

Parameter	**Comments**	**Nominal Values**	**Units**
Effective capacitance, *C*	Input a value normalized by total cell volume as: $\bar{C} = \frac{4C}{A_e \delta N_{pairs}}$.	40-200	F cm^{-3}
Effective resistance, *R*	Input a value multiplied by electrode area: $\bar{R} = R * A_e$. **<u>Note:</u>** the present version of the model does not scale resistance with number of electrodes. As such, it is recommended to modulate the size of the CDI plant by changing A_e and not N_{pairs}.	20-200	Ω cm^2
Cell volume, ∀	Input a value normalized by electrode area: $\bar{\forall} = \frac{\forall}{A_e}$	varies	m
# of Electrode pairs, N_{pairs}	Total electrode pairs in CDI stack.	varies	--
Surface area of an electrode, A_e	Projected surface area of a single electrode.	varies	cm^2
Electrode thickness, δ	Thickness of a single electrode.	0.01-0.05	cm

Internal area of an electrode, a	Internal porous surface area of a single electrode by mass.	500-3000	$m^2\ g^{-1}$
Mass of a single electrode		varies	g
Power recovery efficiency, η	Percentage of energy that is assumed to be recovered from electrodes during regeneration. See Oyarzun et al. (2020)	0-40	%

Table 5. CDI control inputs

Parameter	**Comments**	**Units**
Forcing function	Input a smooth, continuous function of time. Values must be normalized by electrode area: $\overline{I(t)} = \frac{I(t)}{A_e}$. **Note:** please enter "@(t)" at the beginning of the typed equation to allow Matlab to parse this input as a time-varying function. **Note:** see section 1.7 for guidance on specifying constant current operation.	$A\ m^{-2}$
Cycle duration	Time, in seconds, that is required for the forcing function to complete one cycle. For simple sinusoidal forcing functions, for example, the cycle duration is equal to the period of the sinusoid.	s
Voltage gain DC offset	DC component of cell voltage. As the present version of the CDI program only accepts current forcing, the DC component of cell voltage is a free parameter.	V
Voltage threshold	Used for the voltage threshold electrode replacement criterion. Threshold above which electrode oxidation is considered. This value is only used if "run economic analysis" is checked, and the electrode replacement criterion is set to "Hrs_above_V_thresh."	V
Coulombic efficiency, λ_c	Measure of applied current density lost due to leakage currents at higher cell voltages. See Fig. S5 of Hawks et al. (2019) to estimate a value of coulombic efficiency given an expected current density.	--
Differential charge efficiency, $\overline{\lambda_{dl}}$	Cycle-averaged electric double layer differential charge efficiency. See Section S2 of Hasseler et al. (2020) to estimate a value.	--

Table 6. Feed control inputs

Parameter	**Comments**	**Units**
Feed concentration, c_0	Initial concentration of saltwater input into the CDI cell through the feed concentration Simulink block.	mM
Charging /discharging flowrate, *Q(t)*	Input a smooth, continuous function of time. Alternatively, a constant numerical value can also be input. Values must be normalized by electrode area: $\overline{Q(t)} = \frac{Q(t)}{A_e}$. **Note:** please enter "@(t)" at the beginning of the typed equation to allow Matlab to parse this input as a time-varying function (even if a constant value is input).	$\mathrm{L\ min^{-1}\ m^{-2}}$

1.4. Running standalone simulations

Once the Simulink configuration, CDI plant parameters, control parameters, and flow rate parameters have been specified, simple simulations without economic studies can now be run. This operating mode can be performed very quickly and allows the user to verify the operating conditions of the cell as well as calculate all relevant performance parameters before running extended economics studies. Note in this running mode, the *"Economic Analysis"* output tab is not be available.

1. In the *"Sim Parameters"* tab, specify the number of cycles desired to be run.
2. At the bottom of the GUI window, click on the button titled *"Run Model."* **Note:** the first run after initially launching Matlab will take a considerably longer time to execute as Simulink must be launched and initialized.
3. The CDI program will now run and display results in the Model Outputs section of the GUI.

1.5. Running economic studies

1. To run economic studies, check the *"Run Economic Analysis"* box at the top of the *"Economic"* tab in the User Inputs section. Table 7 lists the available economic parameters that can be specified.

Table 7. Economic parameters

Parameter	**Comments**	**Available Options / Values**
Number of years	Desired timespan for the operational life of the CDI system.	(0,inf)
Daily operating schedule	Specify the number of hours the plant will operate per day and the hour of the day that operation will begin.	(0,24]
Operational days per year	Number of days that the CDI cell will run per year.	(0,365]
Discount rate, *i*	Discount rate used in economic present worth of cost analysis. Used to convert future costs / receipts into present values.	[0, inf)
Compounding periods per year	Choose a value of 1 for interest compounded annually.	[0,inf)
Capital costs	Specify the amount of money spent in year 0 to develop and build the CDI plant. To be consistent with other levelized cost sources, the capital cost must be a unit cost on a per surface area of electrodes basis [$ m^{-2}].	[0,inf)
Electrode replacement criterion	**Years**: replace electrodes every X years. **Cycles:** replace electrodes every X cycles. Must be integer multiples. **Cell volumes:** replace electrodes after X cell volumes worth of water has been flowed through the cell. **Hours above threshold:** replace electrodes after X operational hours have been spent where the cell voltage is above a given voltage threshold. Specify the voltage threshold in the "CDI Control" tab.	-Years -Cycles -Cell volumes -Hours above thresh

2. After all economic parameters have been specified, click *"Run Model"* at the bottom of the GUI window. The program will run for the specified number of years. A text prompt will indicate the progress on the analysis.
3. Once the model has finished running, all relevant economic calculations can be found in the *"Economic Analysis"* tab of the model outputs section.

Note: Depending on the desired length of the economic study, running economic analysis may take an extended amount of time.

1.6. Running in batch mode

In the "*Batch Mode"* tab, users can specify numerical vectors over which to run up to three variables. The system will loop over each specified variable and save a batch output file with a desired filename.

1. Check the *"Run Batch Mode"* box.
2. Enter a desired filename for the saved data file. Note, it is recommended to specify a ".mat" filetype, for example: *"Test001.mat"*. This file will be saved to the current Matlab path.
3. Specify numerical arrays for each desired batch variable. Values are entered in the same way that arrays/vectors are specified in Matlab code. For example: "[0:1:10]", "linspace(0,10,10)", and "[1 2 3 4 5 6 7 8 9 10]", are all valid specifications.
4. If extra batch variables are not needed, specify each as "[1]."
5. For each desired batch variable, simply replace the numerical value for that variable in the inputs section with its corresponding batch variable name. For example, if you want to iterate over cell capacitances, you can replace "44" in the "Capacitance" entry with "B1", where B1 is defined as a vector to be "[44 55 66 77]". See Fig. 2 for an example of the above procedure.
6. Click "Run Model" to execute the batch run. A text prompt will display progress within the run.

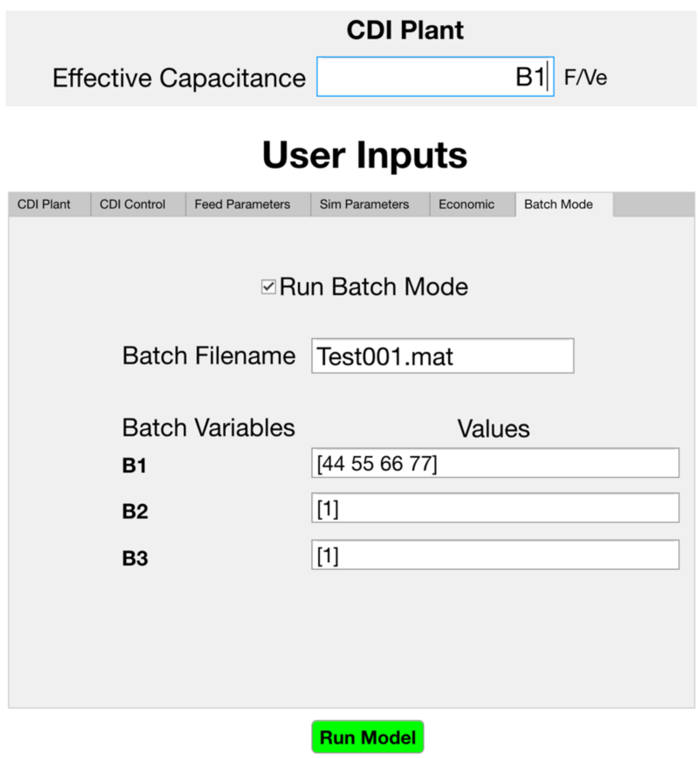

Fig. 2. Sample inputs for running a variable in batch mode

Note: the CDI Program iterates over every permutation of batch variables B1, B2, and B3.

1.7. Running in constant current (CC) mode

To explicitly apply constant current (CC) electrical forcing, specify a two-element column vector in the forcing function box as follows: "@(t) [<*charging current magnitude*> ; <*discharging current magnitude*>]". The program automatically switches polarities after half of a cycle. See Figure 3 for an example input.

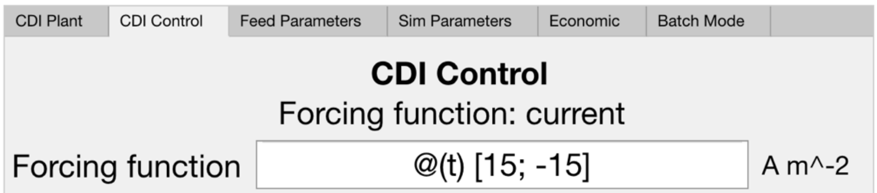

Figure 3. Sample input to specify constant current electrical forcing

2. Glossary of workspace variables

Presented here is a description of the variables found in the Matlab workspace.

Note: The CDI Program converts parameters from the GUI to standard SI base units where applicable. Because of this, values saved in the Matlab workspace may have different units than those input by the user in the GUI.

2.1. ModelParams

These variables are input/calculated from inputs given in the GUI.

`number_of_CDI_electrodes`
- *Description:* total number of electrodes in the CDI stack: $N_{pairs} * 2$
- *Units:* n/a

`A_CDI_electrode`
- *Description:* projected surface area of a single electrode, A_e
- *Units:* m^2

`electrode_thickness`
- *Description:* thickness of a single electrode, δ
- *Units:* m

`R`
- *Description:* effective resistance, R
- *Units:* Ω

`C`
- *Description:* effective capacitance, C
- *Units:* F

`V`
- *Description:* cell volume, $\forall$
- *Units:* m^3

`mass_electrode`
- *Description:* mass of a single electrode
- *Units:* g

`a`
- *Description:* internal porous surface area of a single electrode, a
- *Units:* m^2

`eta_power_recovery`
- *Description:* power recovery efficiency, η
- *Units:* unitless

`lambda_c`
- *Description:* Coulombic efficiency, λ_c
- *Units:* unitless

`lambda_dl`
- *Description:* differential charge efficiency, $\overline{\lambda_{dl}}$
- *Units:* unitless

`Lambda_hat`

- *Description:* effective dynamic charge efficiency, $\overline{\Lambda} \equiv \lambda_c \cdot \overline{\lambda_{dl}}$
- *Units:* unitless

`tau`

- *Description:* flow residence timescale, $\tau \equiv \forall/Q$
- *Units:* s

`V_offset`

- *Description:* cell voltage DC offset
- *Units:* V

`V_thresh`

- *Description:* cell voltage threshold used for the voltage-based electrode replacement criterion
- *Units:* V

2.2. SimulationOutputs

These variables are raw outputs from the Simulink model. Many are used as intermediate values for calculating performance parameters. As such, many of the variables presented here are of limited use but are explained for completeness.

`Brine/Desal_Vol`

- *Description:* timeseries of cumulative wastewater/product-water production versus simulation time.
- *Units:* m^3

`Delta_ND`

- *Description:* timeseries of cumulative moles of salt removed versus simulation time. Reset each cycle. Used to calculate Δc.
- *Units:* mol

`Delta_c_Cyc`

- *Description:* value of volume-averaged desalination depth, Δc. Taken at the end of a cycle. This is the value of the performance parameter that is displayed in the GUI.
- *Units:* mM

`DisposalCost`

- *Description:* timeseries of cumulative wastewater disposal cost versus simulation time.
- *Units:* $

`E_V`

- *Description:* timeseries of volumetric energy consumption calculation. The respective performance parameter is calculated by taking the timeseries value at the end of a cycle. Reset each cycle.
- *Units:* kWh m^{-3}

`E_v_Cyc`

- *Description:* value of volumetric energy consumption calculation, E_v. Taken at the end of a cycle. This is the value of the performance parameter that is displayed in the GUI.
- *Units:* kWh m^{-3}

`ElectrodeCost`

- *Description:* timeseries of electrode cost. Used to transmit electrode cost from Simulink model to the GUI.
- *Units:* $

`Forcing_Current`

- *Description:* timeseries of electrical forcing input.
- *Units:* A

`Forcing_Voltage`

- *Description:* timeseries of cell voltage.
- *Units:* V

`GridEnergyCost`

- *Description:* timeseries of cumulative costs from grid energy block.
- *Units:* $

`LaborCost`

- *Description:* timeseries of costs from labor cost block. Expressed as a percentage of Capital Costs. Used to transmit labor cost factor from Simulink model to the GUI.
- *Units:* %

`Prod`

- *Description:* timeseries of productivity calculation. The respective performance parameter is calculated by taking the timeseries value at the end of a cycle. Reset each cycle.
- *Units:* $\mathrm{L\ m^{-2}\ h^{-1}}$

`Prod_Cyc`

- *Description:* value of productivity calculation, *P*. Taken at the end of a cycle. This is the value of the performance parameter that is displayed in the GUI.
- *Units:* $\mathrm{L\ m^{-2}\ h^{-1}}$

`Q_Brine/Desal`

- *Description:* timeseries of waste/desalination water flow rate.
- *Units:* $\mathrm{m^3\ s^{-1}}$

`VthreshTime`

- *Description:* timeseries of cumulative time spent above the voltage threshold for the voltage-based electrode replacement criterion.
- *Units:* s

`WR`

- *Description:* timeseries of water recovery ratio calculation. The respective performance parameter is calculated by taking the timeseries value at the end of a cycle. Reset each cycle.
- *Units:* unitless

`WR_Cyc`

- *Description:* value of water recovery ratio calculation, *WR*. Taken at the end of a cycle. This is the value of the performance parameter that is displayed in the GUI.
- *Units:* unitless

`c`

- *Description:* timeseries of instantaneous effluent concentration, *c(t)*
- *Units:* mM

`delta_c_average`

- *Description:* timeseries of volume-averaged desalination depth, Δc. The respective performance parameter is calculated by taking the timeseries value at the end of a cycle. Reset each cycle.
- *Units:* mM

`delta_c_instant`

- *Description:* instantaneous desalination depth, $\Delta c(t)$. Not to be confused with the volume-averaged performance parameter.
- *Units:* mM

`tout`

- *Description:* vector of Simulink simulation times. Equivalent to `Tvec`.
- *Units:* s

2.3. SimulinkInputs

These variables are saved to the Matlab workspace so that Simulink can read them.

`c0`

- *Description:* feed concentration, c_0
- *Units:* mM

`Tcyc`

- *Description:* time of one complete electrical forcing cycle
- *Units:* s

`tfinal`

- *Description:* final Simulink simulation time
- *Units:* s

`t0`

- *Description:* initial Simulink simulation time
- *Units:* s

`Tvec`

- *Description:* vector of Simulink simulation times. Fixed to be: [`t0:1:tfinal`]
- *Units:* s

`charging/discharging_flowrate_timeseries`

- *Description:* timeseries of flowrate inputs for charging/discharging. Used for water flow block lookup tables in Simulink.
- *Units:* m^3 s^{-1}

`tauc/taudc_timeseries`

- *Description:* timeseries of flow residence time scale for charging/discharging: $\tau(t) = \forall/Q(t)$
- *Units:* s

`forcing_function_timeseries`

- *Description:* timeseries of current forcing input. Used for lookup table in Simulink.
- *Units:* A

`Forcing_function_IC`

- *Description:* forcing function evaluated at `t0`. Used for RC circuit calculation in Simulink.
- *Units:* A

`day`

- *Description:* day of the year, used for economic calculations. $day \in [1,365]$.
- *Units:* n/a

2.4. EconomicOutputs

These variables represent select outputs from the economic analysis section.

`CAPX_present_levelized`

- *Description:* present worth of capital expenditures. Levelized by the total volume of desalinated water produced over the plant lifetime.
- *Units:* $ m^{-3}

`CycleAveragedCostsPlotData`

- *Description:* plot data corresponding to the plot titled "cycle averaged costs" in the economic outputs section.
- *Units:* $ m^{-3}

`OPX_present_levelized`

- *Description:* present worth of lifetime operating expenditures. Levelized by the total volume of desalinated water produced over the plant lifetime. Represents the contributions of energy, disposal, electrode replacements, and labor/maintenance.
- *Units:* $ m^{-3}

`CostBreakdownPlotData`

- *Description:* plot data corresponding to the plot titled "cost breakdown" in the economic outputs section.
- *Units:* $ m^{-3}

`TotalLevelizedCostPlotData`

- *Description:* plot data corresponding to the plot titled "total levelized cost of water" in the economic outputs section.
- *Units:* $ m^{-3}

`LC_present_levelized`

- *Description:* total present worth of the levelized cost of desalinated water. Levelized by the total volume of desalinated water produced over the plant lifetime. Represents the sum of `CAPX_present_levelized` and `OPX_present_levelized`.
- *Units:* $ m^{-3}

2.5. Batch Mode Outputs

Batch mode outputs are organized in $N \times M \times K$ arrays, where N is the length of batch variable 1 (B1), M is the length of batch variable 2 (B2), K is the length of batch variable 3 (B3). For example, to reference the resulting value of Water Recovery for the n^{th} value of B1, the m^{th} value of B2, and the k^{th} value of B3, index in Matlab as:

```
>> WRBatch(n,m,k)
```

Below is a summary of each variable saved in the batch mode output file:

`batchVar`
- *Description:* Vector of values that each batch variable takes
- *Units:* varies

`DeltaCBatch`
- *Description:* Array of volume-averaged desalination depths (Δc) for each batch run, taken at the end of a cycle.
- *Units:* mM

`ElectrodesPerYearBatch`
- *Description:* Array of number of electrode replacements per year, on average.
- *Units:* y^{-1}

`EVBatch`
- *Description:* Array of volumetric energy consumption (E_v) for each batch run, taken at the end of a cycle.
- *Units:* kWh m^{-3}

`LCBatch`
- *Description:* Array of levelized cost of water (LC) for each batch run.
- *Units:* \$ m^{-3}

`NcycBatch`
- *Description:* Array of number of cycles run per day for each batch run.
- *Units:* day

`ProdBatch`
- *Description:* Array of productivity (*P*) for each batch run, taken at the end of a cycle.
- *Units:* L m^{-2} h^{-1}

`VMaxBatch/VMinBatch`
- *Description:* Array of maximum/minimum resulting cell voltage achieved for each batch run.
- *Units:* V

`WaterProducedBatch`
- *Description:* Array of amount of water produced during desalination phase per day on average, for each batch run.
- *Units:* L day^{-1}

`WRBatch`
- *Description:* Array of water recovery ratio (*WR*) for each batch run, taken at the end of a cycle.
- *Units:* unitless

3. References

Hasseler, T.D., Ramachandran, A., Tarpeh, W.A., Stadermann, M., Santiago, J.G., 2020. Process design tools and techno-economic analysis for capacitive deionization. Water Res. https://doi.org/10.1016/j.watres.2020.116034

Hawks, S.A., Ramachandran, A., Porada, S., Campbell, P.G., Suss, M.E., Biesheuvel, P.M., Santiago, J.G., Stadermann, M., 2019. Performance metrics for the objective assessment of capacitive deionization systems. Water Res. https://doi.org/10.1016/j.watres.2018.10.074

Oyarzun, D.I., Hawks, S.A., Campbell, P.G., Hemmatifar, A., Krishna, A., Santiago, J.G., Stadermann, M., 2020. Energy transfer for storage or recovery in capacitive deionization using a DC-DC converter. J. Power Sources. https://doi.org/10.1016/j.jpowsour.2019.227409