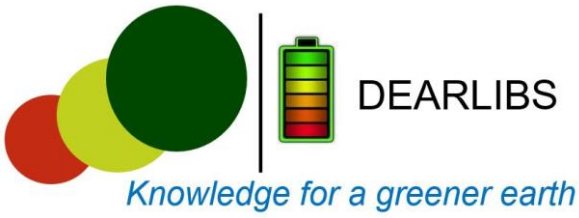
DEARLIBS



A Robust and Sleek Electrochemical Battery Model Implementation: a MATLAB® Framework

Developers: Seongbeom Lee[1] and Simona Onori[2]

Department of Energy Resources Engineering,
Stanford University, Stanford, CA 94305, USA

Manual

We provide **DEARLIBS: D**oyle-Fuller-Newman **E**lectrochemical B**a**ttery Model Implementation in **R**obust and Sleek MATLAB® Framework for **Li**thium-ion **B**atterie**s**. Users can freely download the MATLAB® framework from https://github.com/DEARLIBS. Users should download an m-file (P2D_identificaiton.m) and a text file (voltage_exp.txt) in the same folder. 'P2D_identifcation.m' consists of differential algebraic equations (DAEs) after numerical discretization, a standard MATLAB® solver (*ode15s*), a parameter identification routine along with voltage data from experimental measurements (voltage_exp.txt). DEARLIBS adopts the finite difference method, over which the single-step iteration-free initialization approach is applied. The objective of DEARLIBS is to provide a MATLAB® framework for the robust and sleek implementation of physics-based battery models, where the DFN model is adopted as an example (note: 'robust' in this context refers to the ability of the solver to generate a successful simulation without a crash- e.g., even under extreme fast charging and/or dynamic battery operation).

1. MATLAB® installation

To implement DEARLIBS, MATLAB® is required.
Users can find detailed installation instructions in the following MATLAB® webpage:
https://www.mathworks.com/help/install/install-products.html

2. Required Toolbox

Symbolic and global optimization toolboxes are required to execute DEARLIBS. The parallel computing and signal processing toolboxes are optional: (i) The '*parfor*' command in the MATLAB® code can be replaced with the '*for*' command if users do not have a parallel computing toolbox. The '*parfor*' command only accelerates computational time and is used on Lines 105, 112, 123, 134, 145, 154, 167, 191, 208, 223, 239, 248, 260, 268, 288, 308, 332, 346, 364, 389, 393, 397, 404, 407, 413, 416, 423, 428, 436, 440, 444 in the MATLAB® code; (ii) If users do not have a signal processing toolbox, they can directly write down the root-mean-

[1] sblee3487@gmail.com
[2] sonori@stanford.edu

square equation instead of the ‘rms’ command provided in the MATLAB® code. The ‘rms’ command is used on Lines 569 in the MATLAB® code.

3. Instruction

1) Users can input the applied C-rate (Line 25 in the MATLAB® code). The negative sign represents discharge, and the positive sign represent charge.

```
22 -    global N M NM f MM y0
23
24      %% C-rate (Put your C-rate)
25 -    Crate=-1;
26
```

2) Users need to input total discharging time and numbers of experimental data (Line 28-29). In DEARLIBS, total 63 experimental voltage data with the 50 seconds interval for 3100 seconds for discharging time are adopted (see voltage_exp.txt).

```
26
27      %% Experimental data (Users can input their experimental conditions)
28 -    Numexp=63;                          % Number of experimental data
29 -    Totexp=3100;                        % Experimental discharging time (or charging time)
```

3) User can input numbers of parameters to be identified (Line 32 in the MATLAB® code).

```
30
31      %% Number of parameters to be identified (Input the number of parameters to be identified)
32 -    n_vars = 8;                        % Number of parameters to be identified
33 -    kk = sym('kk',[1 n_vars]);         % Arrays for parameters to be identified
34
```

4) Users can input numbers of node points (Line 36~38 in the MATLAB® code).

```
34
35      %% Node number (Change # of node- N: Cathode, M: Membrance, NM: Cathode) (Put your number of node points)
36 -    N=2;
37 -    M=2;
38 -    NM=2;
```

5) Users can input model parameters (Line 40~83 in the MATLAB® code). In this part, users can determine parameters to be identified. Identified parameters are expressed as kk(i), i=1..n. In this code, transport and kinetic model parameters are identified.

```
39
40      %% Design Parameters
41 -    ep=0.335;                  % Porosity at positive
42 -    es=0.47;                   % Porosity at membrane
43 -    en=0.25;                   % Porosity at negative
44 -    brugp=2.43;                % Bruggeman coefficient at positive
45 -    brugs=2.57;                % Bruggeman coefficient at separator
46 -    brugn=2.91;                % Bruggeman coefficient at negative
47 -    lp=75.6e-6;                % Thickness at positive (unit:m)
48 -    ls=12e-6;                  % Thickness at membrance (unit:m)
49 -    lnl=85.2e-6;               % Thickness at negative (unit:m)
50 -    Rpp=5.22e-6;               % Radius of solid particle at positive (unit:m)
51 -    Rpn=5.86e-6;               % Radius of solid particle at negative (unit:m)
52 -    F=96487;                   % Faraday constant (unit: C/mol)
53 -    R=8.3143;                  % Ideal gas constant (unit: J/(mol K))
54 -    tl=0.363;                  % Transference coefficient
55 -    ap=(3/Rpp)*(1-ep);         % Particle surface area to volume at positive (unit: m^2/m^3)
56 -    an=(3/Rpn)*(1-en);         % Particle surface area to volume at negative (unit: m^2/m^3)
57 -    T=298.15;                  % Temperature (K)
58 -    Acell=0.11;                % Electrode area (m^2)
59 -    Capa=5;                    % Nominal capacity (Ah)
60 -    iapp=Capa*Crate/Acell;     % Applied current density (A/m^2)
61
```

```
61
62      %% Transport parameters
63 -    c0=1000;                     % Electrolyte concentration (unit:mol/m3)
64 -    Dl=kk(1)*10^(-9);            % Electrolyte diffusion coefficient (unit:m^2/s)
65 -    Kappa=kk(2);                 % Conductivity (unit:S/m)
66 -    ctp=51765;                   % Maximum solid phase concentration at positive (unit:mol/m^3)
67 -    ctn=29583;                   % Maximum solid phase concentration at negative (unit:mol/m^3)
68 -    Dbulk=Dl;                    % Electrolyte diffusivity (unit:m^2/s)
69 -    sigmap=kk(3);                % Solid phase conductivity at positive (unit:S/m)
70 -    sigman=kk(4);                % Solid phase conductivity at negative (unit:S/m)
71 -    Dsp=kk(5)*10^(-15);          % Solid particle diffusivity at positive (unit:m^2/s)
72 -    Dsn=kk(6)*10^(-14);          % Solid particle diffusivity at negative (unit:m^2/s)
73
74 -    Keffp=Kappa*(ep^brugp);      % Liquid phase conductivity at positive (unit:S/m)
75 -    Keffs=Kappa*(es^brugs);      % Liquid phase conductivity at membrane (unit:S/m)
76 -    Keffn=Kappa*(en^brugn);      % Liquid phase conductivity at negative (unit:S/m)
77 -    D2pos=(ep^brugp)*Dbulk;      % Electrolyte diffusivity at positive (unit:m^2/s)
78 -    D2sep=(es^brugs)*Dbulk;      % Electrolyte diffusivity at membrane (unit:m^2/s)
79 -    D2neg=(en^brugn)*Dbulk;      % Electrolyte diffusivity at negative (unit:m^2/s)
80
81      %% Kinetic parameters
82 -    kp=kk(7)*10^(-11);           % Reaction rate constant at positive (unit:m^2.5/(mol^0.5 s))
83 -    kn=kk(8)*10^(-12);           % Reaction rate constant at negative (unit:m^2.5/(mol^0.5 s))
```

6) The open-circuit voltages at the positive and negative electrodes are introduced in Line 157 and 170 in the MATLAB® code, respectively.

6-1) Users can input the open-circuit voltage (U_p) at the positive electrode depending on their lithium-ion chemistries.

```
153 -    jp=sym(zeros(1,N+1));
154 -    parfor i=2:N+1
155
156 -    theta=u2(i)*ctp/ctp;
157 -    Up=-0.8090*theta+4.4875-0.0428*tanh(18.5138*(theta-0.5542))-17.7326*tanh(15.7890*(theta-0.3117))+17.5842*tanh(15.9308*(theta-0.3120));
```

6-2) Users can input the open-circuit voltage (U_n) at the positive electrode depending on their lithium-ion chemistries.

```
parfor i=1+N+1+M+1+1:1+N+1+M+1+NM

theta=u2(i)*ctn/ctn;
Un=1.9793*exp(-39.3631*theta)+0.2482-0.0909*tanh(29.8538*(theta-0.1234))-0.04478*tanh(14.9159*(theta-0.2769))-0.0205*tanh(30.4444*(theta-0.6103));
```

7) In this code, governing equations of the P2D model are discretized by the finite difference method to obtain ordinary differential and algebraic equations. The ordinary differential and algebraic equations are directly coded in a symbolic form (see Line 176~376). In addition, the single-step iteration-free initialization method is applied to discretized differential algebraic equations (see Line 378~451).

8) Users can input initial guesses for electrolyte concentration, solid-phase surface concentration at the positive and negative electrode, solid-phase average concentration at the positive and negative electrode, solid-phase potential at the positive and negative electrode, a liquid phase potential (Line 458~466 in the MATLAB® code).

```
%% Initial condition guess (Put your initial guess)
U(1:1+N+1+M+1+NM+1)=1;                                                    % Electrolyte concentration
U(1+1+N+1+M+1+NM+1:N+1+N+1+M+1+NM+1)=.27;                                 % Surface con. at positive
U(1+1+N+1+M+1+NM+1+N:NM+1+N+1+M+1+NM+1+N)=0.9014;                         % Surface con. at negative
U(1+1+N+1+M+1+NM+1+N+NM:N+1+N+1+M+1+NM+1+N+NM)=.27;                       % Average con. at positive
U(1+1+N+1+M+1+NM+1+N+NM+N:NM+1+N+1+M+1+NM+1+N+NM+N)=0.9014;               % Average con. at negative
U(1+1+N+1+M+1+NM+1+N+NM+N+NM:N+2+1+N+1+M+1+NM+1+N+NM+N+NM)=4.30430037;    % Solid phase potential at positive
U(1+1+N+1+M+1+NM+1+N+NM+N+NM+N+2:NM+2+1+N+1+M+1+NM+1+N+NM+N+NM+N+2)=.9202000152e-1;  % Solid phase potential at negative
U(1+1+N+1+M+1+NM+1+N+NM+N+NM+N+2+NM+2:1+N+1+M+1+NM+1+1+N+1+M+1+NM+1+N+NM+N+NM+N+2+NM+2)=0;  % Liquid potential

y0 = (U(1:1+N+1+M+1+NM+1+1+N+1+M+1+NM+1+N+NM+N+NM+N+2+NM+2));
```

9) Experimental voltage measurement is loaded (Line 470 in the MATLAB® code). Users can also load their experimental voltage data.

```
x=importdata('voltage_exp.txt');
```

10) In the DEARLIBS, the PSO algorithm is implemented (Line 472~482 in the MATLAB® code). Users can input initial guesses and upper/lower bounds for model parameters to be identified.

```
%% PSO identificaiton (Input your uppoer and lower bounds for parameters to be identified. Input your options for PSO)

pp=0.3;                 % Deviation for upper and lower bounds (percentage)
D10=1;                  % Electrolyte diffusion coefficient
Kappa0=1.17;            % Conductivity
sigmap0=0.18;           % Solid phase conductivity at positive
sigman0=215;            % Solid phase conductivity at negative
Dsp0=4;                 % Solid particle diffusivity at positive
Dsn0=3.3;               % Solid particle diffusivity at negative
kp0=0.7;                % Reaction rate constant at positive
kn0=0.7;                % Reaction rate constant at negative
```

11) Users can input the '*ode15'* solver option (Line 515-518 and 561-565).

```
515 -    opt = odeset('Mass', M1,'MStateDependence','weak','RelTol',1e-5,'AbsTol',1e-5,'InitialStep',1e-3,'MaxStep',5,'Events',@stopcondition);
516 -    warning('off','all');
517 -    tic
518 -    [T,Y] = ode15s(F,[0 tsp], y0, opt);
```

12) Users can execute DEARLIBS. Users will have a comparison of voltage profiles between modeling outputs and experimental data, computational time, identified parameter values, and a root-mean-square error as following:

```
>> P2D_identification

                                 Best            Mean     Stall
Iteration     f-count            f(x)            f(x)    Iterations
    0              10         0.02192         0.02949        0
    1              20         0.02192         0.04029        0
    2              30         0.02186         0.03568        0
    3              40         0.02186         0.03317        1
    4              50         0.02186         0.03289        2
    5              60         0.02186         0.03037        3
    6              70         0.02149         0.03626        0
    7              80         0.02149         0.03009        1
    8              90         0.02149          0.0415        2
    9             100         0.02149         0.03223        3
   10             110         0.02149         0.04032        4
Optimization ended: number of iterations exceeded OPTIONS.MaxIterations.
Elapsed time is 48.866400 seconds.
Elapsed time is 0.492657 seconds.
```

identified_params_kinetic.mat (MAT-file)

Name	Value
D1	1.1372e-09
Dsn	4.2900e-14
Dsp	4.1931e-15
Kappa	1.0796
fval	0.0215
kn	6.1965e-13
kp	8.2032e-12
sigman	279.5000
sigmap	0.1904