Instructions file to run the program and generate the results discussed in the article

Tanks-in-Series MATLAB Code Dissemination

Note: All the results discussed in the article is based on the executable generated using SUNDIALS-IDA C-program DAE solver. It is an executable file that reads the parameters from the external text file. The chemistry of the model can't be changed in executable file. The code dissemination of this work is shared based on the TiS model simulation performed using MATLAB ODE15s solver. The MATLAB code has facility to change the OCV data/expressions, parameters, and other constitutive expressions based on the chemistry of the cell under the study.

In the directory, "**Code_Dissemination_MATLAB**" there are two sub-directories:

Sub-Directory: Figure3_ReferenceData

1. In this directory, the synthetic experimental data is generated using M-file "**Main_SynData_Gen_p2D.m**". "**p2Dmodel.exe**" is the p2D model simulation executable that reads p2D_ippars.txt.
2. The mentioned M-file generates a mat file "synthetic_p2D_data.mat". It is the synthetic experimental data generated by p2D model simulation

Sub-Directory: TiS_ModelSimulation_ParamEstim

In this directory, MATLAB code demonstrates discharge simulation and parameter estimation using GA. ODE15s tool is used to simulate Differential Algebraic Equations (DAEs).

1. "**Main_LiTank_1_DischargeSimulation.m**" is the M-file that should be used to perform discharge simulation.
2. "**Main_LiTank_2_GA_ParamEstimation.m**" is the main M-file that performs parameter estimation using GA (genetic algorithm toolbox).
3. "**LiTank_Dyn.m**" is the function file that has Tanks-in-Series model equations. It also has sub-function files that enables to change constitutive relations such as electrolyte

diffusivities, ionic conductivity and open circuit potential relations for positive and negative electrodes.

4. "**LiTank_ICs.m**" is the function file that reads initial conditions.
5. "**LiTank_Parameters.m**" is the function file that read list of parameters required to simulate the model.
6. "**stop_condition_cc.m**" is the function file required to specify the stop condition to clip the simulation whenever the simulation hits maximum or minimum cell potential.
7. "**Estimation_Results_Plots.m**" is the function file that plots after parameter estimation (optimization).
8. "**calc_rmse.m**" is the function file that calculates root mean square error (RMSE) between experimental data and model simulation based on the cell potential.
9. "**TankModel_ObjFcn.m**" is the function file that computes objective function to estimate parameters. ODE15s is used to simulate the DAEs of the TiS model, followed by calculate RMSE for C/2 and 1C-rate.

Tanks-in-Series Model Simulation in MATLAB

The DAE models are written in the following form

$$\begin{aligned} \frac{dy}{dt} &= f\left(y,z\right) \\ 0 &= g\left(y,z\right) \end{aligned} \tag{1}$$

Where y is the variable mappled to the differential equations and z is the variable mapped to the algebraic equations. This system of equations can be solved using DAE solver in time such as ODE15s in MATLAB. However, because of varying dynamic systems and multiscale nature of different time constants, the adaptive solver (ODE15s) in MATLAB cannot simulate till the cut-off potential. For MATLAB demo, we adpated the effective scaling methodology proposed by our group in the article *J.Electrochem. Soc.*, **167**, 163503 (2020). In TiS for lithium-ion battery model, we define log scaling of electrolyte potential as $\overline{\phi_{l,i}} = \exp\left(\phi_{l,i}\right)$. For convenience, the log scaled electrolyte variable is defined as $\ln\left(\overline{\phi_{l,i}}\right) = \phi_{l,i}$. The final set of TiS model equations are given in Table I. Refer Table II and Table IV for the constitutive expressions, parameters.

Table I. Governing equations of the Tanks-in-Series Model.

Positive Electrode (Region 1)	Separator (Region 2)	Negative Electrode (Region 3)
$\dfrac{d\overline{c_1}}{dt}=\dfrac{\dfrac{2D_{elec}(\overline{c_2}-\overline{c_1})}{\dfrac{l_1}{\varepsilon_1^{b_1}}+\dfrac{l_2}{\varepsilon_2^{b_2}}}}{\varepsilon_1 l_1}+\left(1-t_+^0\right)\dfrac{i_{app}}{F\varepsilon_1 l_1}$ $i_{app}=-2\kappa(c_{12})\left(\dfrac{\phi_{l,2}-\phi_{l,1}}{\dfrac{l_1}{\varepsilon_1^{b_1}}+\dfrac{l_2}{\varepsilon_2^{b_2}}}\right)+\dfrac{4RT(1-t_+^0)}{F}\nu(c_{12})\kappa(c_{12})\dfrac{1}{c_{12}}\left(\dfrac{\overline{c_2}-\overline{c_1}}{\dfrac{l_1}{\varepsilon_1^{b_1}}+\dfrac{l_2}{\varepsilon_2^{b_2}}}\right)$ $\dfrac{d\overline{c_1^{s,avg}}}{dt}=-3\dfrac{\overline{j_1}}{R_1}$ $\dfrac{D_1^s}{R_1}[\overline{c_1^{s,surf}}-\overline{c_1^{s,avg}}]=-\dfrac{\overline{j_1}}{5}$ $\dfrac{i_{app}}{Fa_1 l_1}=k_1\left(\overline{c_1}\right)^{\alpha_{c,1}}{}_1(c_1^{s,\max}-\overline{c_1^{s,surf}})^{\alpha_{a,1}}(\overline{c_1^{s,surf}})^{\alpha_{c,1}}\left(\exp\left(\dfrac{\alpha_{a,1}F\overline{\eta_1}}{RT}\right)-\exp\left(\dfrac{-\alpha_{c,1}F\overline{\eta_1}}{RT}\right)\right)$ $\overline{\eta_1}=\overline{\phi_{s,1}}-\phi_{l,1}-U(\overline{c_1^{s,surf}})$ $c_{12}=\left(\dfrac{\dfrac{\varepsilon_1^{b_1}}{l_1}\overline{c_1}+\dfrac{\varepsilon_2^{b_2}}{l_2}\overline{c_2}}{\dfrac{\varepsilon_1^{b_1}}{l_1}+\dfrac{\varepsilon_2^{b_2}}{l_2}}\right)$ $\nu\left(c_{12}\right)=1+\dfrac{\partial\ln f}{\partial\ln c_{12}}$	$\dfrac{d\overline{c_2}}{dt}=\dfrac{\dfrac{-2D_{elec}(\overline{c_2}-\overline{c_1})}{\dfrac{l_1}{\varepsilon_1^{b_1}}+\dfrac{l_2}{\varepsilon_2^{b_2}}}+\dfrac{2D\left(c_{23}\right)(\overline{c_3}-\overline{c_2})}{\dfrac{l_2}{\varepsilon_2^{b_2}}+\dfrac{l_3}{\varepsilon_3^{b_3}}}}{\varepsilon_2 l_2}$ $\phi_{l,12}=\left(\dfrac{\dfrac{\varepsilon_1^{b_1}}{l_1}\phi_{l,1}+\dfrac{\varepsilon_2^{b_2}}{l_2}\phi_{l,2}}{\dfrac{\varepsilon_1^{b_1}}{l_1}+\dfrac{\varepsilon_2^{b_2}}{l_2}}\right)=0$	$\dfrac{d\overline{c_3}}{dt}=\dfrac{\dfrac{-2D_{elec}(\overline{c_3}-\overline{c_2})}{\dfrac{l_2}{\varepsilon_2^{b_2}}+\dfrac{l_3}{\varepsilon_3^{b_3}}}}{\varepsilon_3 l_3}-\left(1-t_+^0\right)\dfrac{i_{app}}{\varepsilon_3 l_3}$ $i_{app}=-2\kappa(c_{23})\left(\dfrac{\phi_{l,3}-\phi_{l,2}}{\dfrac{l_3}{\varepsilon_3^{b_3}}+\dfrac{l_2}{\varepsilon_2^{b_2}}}\right)+\dfrac{4RT(1-t_+^0)}{F}\nu(c_{23})\kappa(c_{23})\dfrac{1}{c_{23}}\left(\dfrac{\overline{c_3}-\overline{c_2}}{\dfrac{l_2}{\varepsilon_2^{b_2}}+\dfrac{l_3}{\varepsilon_3^{b_3}}}\right)$ $\dfrac{d\overline{c_3^{s,avg}}}{dt}=-3\dfrac{\overline{j_3}}{R_3}$ $\dfrac{D_3^s}{R_1}[\overline{c_3^{s,surf}}-\overline{c_3^{s,avg}}]=-\dfrac{\overline{j_3}}{5}$ $\dfrac{-i_{app}}{Fa_3 l_3}=k_3\left(\overline{c_3}\right)^{\alpha_{a,3}}(c_3^{s,\max}-\overline{c_3^{s,surf}})^{\alpha_{a,3}}(\overline{c_3^{s,surf}})^{\alpha_{c,3}}\left(\exp\left(\dfrac{\alpha_{a,3}F\overline{\eta_3}}{RT}\right)-\exp\left(\dfrac{-\alpha_{c,3}F\overline{\eta_3}}{RT}\right)\right)$ $\overline{\eta_3}=\overline{\phi_{s,3}}-\phi_{l,3}-U(\overline{c_3^{s,surf}})$ $c_{23}=\left(\dfrac{\dfrac{\varepsilon_3^{b_3}}{l_3}\overline{c_3}+\dfrac{\varepsilon_2^{b_2}}{l_2}\overline{c_2}}{\dfrac{\varepsilon_3^{b_3}}{l_3}+\dfrac{\varepsilon_2^{b_2}}{l_2}}\right)$ $\nu\left(c_{23}\right)=1+\dfrac{\partial\ln f}{\partial\ln c_{23}}$