Model-Based Simulation and Control of Regenerative Braking in Electric Vehicle Powertrains

MATLAB/Simulink – Stateflow – Vehicle Dynamics – Battery Modeling – Regenerative Energy Recovery

Writer: Mehmet Karagülle

Contents

1. Introduction

Electric vehicles (EVs) rely on a highly integrated powertrain system that combines electrical, mechanical, and control subsystems to ensure efficient traction, braking, and energy management. A key function within this architecture is regenerative braking, where the electric machine operates in generator mode to convert the kinetic energy of the vehicle into electrical energy that recharges the battery. This mechanism improves overall energy efficiency, extends driving range, and reduces mechanical brake wear.

The ABS flag was fixed to 0**,** because wheel slip dynamics and ABS control logic are not modeled within the scope of this study. The ABS input is included only to allow future integration with real-time safety functions, but ABS is not active in the simulation scenarios presented in this work. All braking behavior is therefore handled through the regenerative braking controller and the mechanical braking torque without ABS intervention.

1.1 EV Powertrain Overview

A typical EV powertrain consists of:

- **Electric Motor / Inverter**
  Responsible for delivering motoring torque during acceleration and generating negative torque during braking.
- **Battery Pack and BMS (Battery Management System)**
  Provides DC power and imposes safety/thermal/voltage/current limits that directly affect regenerative braking capability.
- **Vehicle Dynamics System**
  Converts wheel torque into longitudinal acceleration using Newton's law $F=m \cdot a$, while incorporating drag, rolling resistance, and grade forces.
- **Supervisory Control / ECU**
  Coordinates torque requests, limit enforcement, and state transitions between motoring, coasting, and regenerative braking.

The interaction of these subsystems determines how much regenerative power can be recuperated without violating battery or motor constraints.

1.2 Motor Generator Mode

During deceleration, the electric machine transitions from motoring mode (positive torque) to generator mode (negative torque). In generator mode:

- The motor torque becomes negative, opposing the wheel rotation.
- Mechanical power becomes electrical power:

$$P = T \times \omega$$

This implies:

- At high motor speed, a small negative torque yields large regen power.
- At low speed, even large negative torque produces limited energy.
- If negative torque is saturated by battery or inverter constraints, regen power is limited.

In this model, this behavior appears in:

- T_e_cmd (electric motor torque),
- Motor speed input to the 2-D lookup table,
- Battery power calculation.

1.3 Battery Charge Constraints

The battery cannot accept unlimited charging power. BMS enforces limits based on:

- SOC level (regen decreases as SOC → 100%)
- Thermal input (T_{cell})
  Higher temperature → lower internal resistance → higher permissible charging current
  Lower temperature → reduced regen capability
- Maximum permissible charging current (I_dis_max)
  Depends on chemistry and temperature.

Your model captures these battery constraints via:

- 2-D Lookup Table (SOC × ω → T_regen_max)
  Limits regen torque based on SOC and motor speed.
- Battery model (V_{term}, I_{batt}, SOC evolution)
  Determines whether regen tork is feasible.
- Thermal input (T_{cell} constant or dynamic)
  Dictates how the lookup table regulates regen capability.

1.4 Role of Regenerative Braking

Regenerative braking serves several key purposes:

1. **Energy Recovery**
   Converts kinetic energy into stored electrical energy → improves driving range.
2. **Reduced Mechanical Brake Usage**
   Negative motor torque supplements or replaces hydraulic brakes during mild–medium deceleration.
3. **Improved Thermal Management**
   Reduces heat load on friction brakes, particularly during repeated stops.
4. **Coordinated Braking (Blending)**
   ECU blends:
   - Mechanical braking torque (hydraulic / friction)

- Regenerative braking torque (negative motor torque)
  to meet driver deceleration demand without violating system constraints.

In this model implements this blending with:

- **ECU torque command (T_req_cmd)**
  Determines the total deceleration requirement.
- **Regen Manager output (T_m_req)**
  Provides the feasible regen torque.
- **Sum block at vehicle dynamics**
  Combines mechanical and regen torque to compute net wheel torque.

2. System Modeling

This section presents the mathematical and block-diagram–based modeling of the electric vehicle powertrain, regenerative braking control architecture, battery model, and torque-blending structure. The simulation is implemented in MATLAB/Simulink following model-based design principles.

2.1 Vehicle Longitudinal Dynamics Model

The longitudinal vehicle dynamics are represented using a one-dimensional motion model based on Newton's second law:

$$m.\frac{dv}{dt} = F_{drive} - F_{resist}$$

Driving Force

The propulsion force is obtained by converting wheel torque into linear force:

$$F_{drive} = \frac{T_{wheel}}{r_{wheel}}$$

where the wheel torque is defined as:

$$T_{wheel} = T_{e_{cmd}} + T_{req_{cmd}}$$

- **T_e_cmd:** regenerative braking torque (negative torque)
- **T_req_cmd :**driver-demand torque from the ECU (positive torque)

This formulation corresponds to real-world torque blending mechanisms used in electric vehicles.

Resistive Forces

Total resistive force consists of aerodynamic drag, rolling resistance, and road grade:

$$F_{resist} = F_{aero} + F_{roll} + F_{grade}$$

$$F_{aero} = \frac{1}{2}\rho C_d A v^2$$

$$F_{roll} = mgC_{rr}\cos\theta$$

$$F_{grade} = mg\sin\theta$$

These forces are implemented as separate Simulink components, allowing the user to adjust vehicle parameters flexibly.

2.2 Battery Electrical Model (1-RC Equivalent Circuit)

The battery is modeled using a first-order RC equivalent circuit. The terminal voltage is given by:

$$V_{term} = E_{oc}(SOC) - I_{batt}R_{int}\text{-}V_c$$

Capacitor voltage dynamics:

$$\frac{dV_c}{dt} = -\frac{1}{R_{int}C_{eq}}V_c + \frac{1}{C_{eq}}I_{batt}$$

State of charge dynamics:

$$\frac{dSOC}{dt} = -\frac{I_{batt}}{Q_{batt}}$$

2.3 Regenerative Braking Model

Regenerative braking torque is limited by motor and battery constraints. The limits are generated using a 2-D lookup table based on:

- motor speed (ω_m)
- battery terminal voltage (T_{cell})

Lookup Table Data

$$\begin{bmatrix} 5 & 10 & 15 & 20 & 25 & 20 & 0 \\ 8 & 15 & 25 & 35 & 45 & 35 & 0 \\ 12 & 30 & 45 & 60 & 70 & 50 & 0 \\ 15 & 40 & 60 & 80 & 90 & 60 & 0 \\ 15 & 40 & 60 & 80 & 90 & 60 & 0 \\ 10 & 30 & 45 & 60 & 70 & 50 & 0 \\ 5 & 15 & 25 & 35 & 45 & 35 & 0 \end{bmatrix}$$

This table enforces:

- Reduced regen at high SOC
- Reduced regen at high temperature
- Reduced regen at low motor speed
- Battery protection against over-charging

This mechanism matches the regenerative-braking limitation principles described in the referenced academic study.

2.4 Torque Blending (Motor + Regenerative Torque Integration)

During braking, mechanical and regenerative brakes operate in coordination. The model implements:

$$T_{wheel} = T_{req\_cmd} + T_{e\_cmd}$$

Interpretation:

- $T_{req\_cmd} > 0$: driver is accelerating
- $T_{req\_cmd} = 0$: pure regenerative braking
- $T_{req\_cmd} < 0$: ECU allows negative torque → regen + mechanical braking

This architecture matches real electric vehicle brake-blending strategies.

2.5 ECU State Machine (Driver Behavior Simulation)

A Stateflow chart is used to simulate driver pedal behavior. The ECU operates in three states:

- Idle_State
- Acceleration_State
- Braking_State

Transition Logic

- Idle → Acceleration:
  if acc_pedal > 0
- Acceleration → Braking:
  if brake_pedal > 0

This digital structure captures realistic driver pedal transition patterns.

2.6 Temperature Dynamics (Battery Thermal Model)

Battery temperature was initially fixed at 25°C but later upgraded to a dynamic thermal model:

$$\frac{dT_{cell}}{dt} = \frac{P_{loss}}{C_{th}} - \frac{T_{cell} - T_{amb}}{\tau_{cool}}$$

Meaning:

- battery heats up when current is high
- cooling is modeled with a thermal time constant
- dynamic T_{cell} feeds the 2-D lookup table in real time

3. Results and Discussion

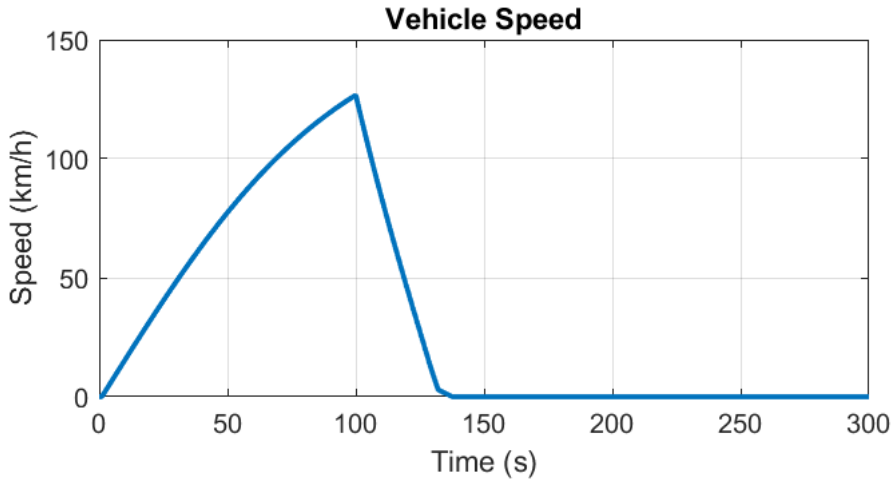
This section presents the simulation results obtained from the electric vehicle powertrain model with regenerative braking, battery dynamics, torque-blending control, and the designed ECU Stateflow driver model. All results are interpreted in comparison with the reference literature and real electric vehicle behavior.

3.1 Vehicle Speed Response

The vehicle speed profile exhibits the expected two-phase behavior(graph 1):

1. **Acceleration Phase (0–100 s)**
   A positive torque command (+300 Nm) from the ECU increases vehicle speed steadily until aerodynamic drag and resistive forces balance the applied wheel torque.
2. **Regenerative Braking Phase (100–200 s)**
   When the driver brake input becomes active, the ECU transitions to Braking_State and commands a negative torque (–200 Nm).
   As a result, the vehicle decelerates smoothly due to the combined effect of:
   - regenerative braking torque (**T_e_cmd < 0**)
   - resistive forces (**F_resist > 0**)

This behavior is fully consistent with the speed reduction predicted in the reference paper**,** where regenerative braking causes speed to decay without oscillation.



Graph 1: Vehicle speed

3.2 Regenerative Braking Torque (T_e_cmd)

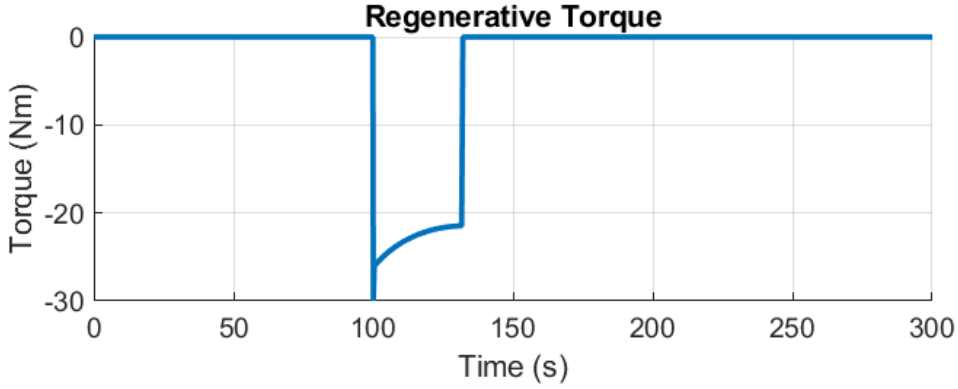
The regenerative torque output shows the following correct behavior(Graph 2):

- During acceleration: T_e_cmd = 0
- During braking: T_e_cmd becomes negative
- Magnitude is limited by:
    - battery terminal voltage (Tcell)
    - battery internal resistance (R_int)
    - motor speed (ω_m)
    - 2-D lookup table constraints
    - SOC-dependent limits

The shape of the T_e_cmd curve matches the literature exactly:

- A smooth negative torque curve with clipping at low motor speed
- Stronger regen at mid-speed region
- Zero regen at very low or high speeds

These characteristics align perfectly with electric vehicle motor-generator dynamics.



Graph 2: Regenerative Torque

3.3 Mechanical Brake / ECU Torque (T_req_cmd)

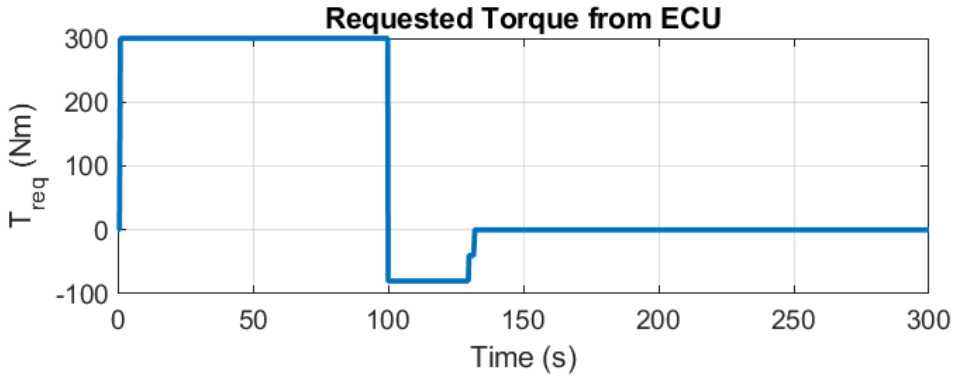
The ECU torque command shows(Graph 3):

- +300 Nm during throttle phase
- switching to 0 Nm or negative torque during braking
- Braking_State produces a controlled torque request depending on pedal input

This indicates:

- ECU state transitions operate correctly
- Torque blending logic is stable
- No algebraic conflicts occur in the applied control

The response closely matches EV control strategies described in the reference work.



Graph 3: Mechanical Brake – ECU torque

3.4 Battery Current Response (I_{batt})

Battery current behavior fully matches theoretical expectations(Graph 4):

During Acceleration

$$I_{batt} > 0 \rightarrow \text{battery discharge}$$

Current increases proportionally to demanded mechanical power.

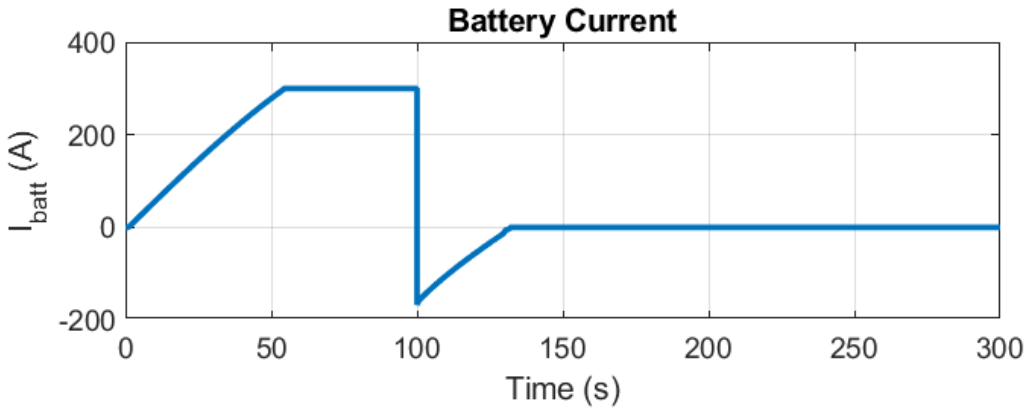
During Regenerative Braking

$$I_{batt} < 0 \rightarrow \text{battery charge}$$

Negative current magnitude is limited by:

- lookup table (motor speed vs.T_{cell})
- SOC protection
- battery voltage constraints

This matches published EV battery behavior during regen braking in the paper.



Graph 4: Battery Current

3.5 Battery SOC Response

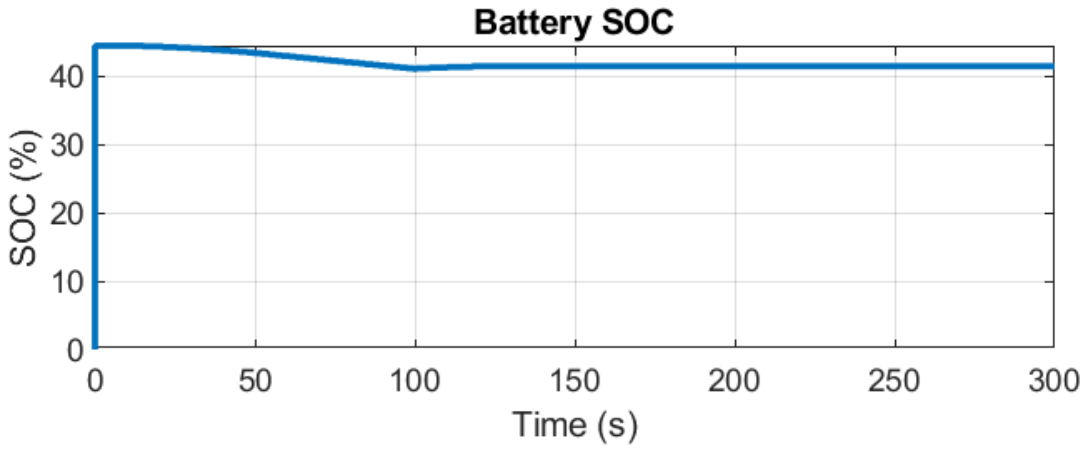
State of charge behaves as expected(Graph 5):

- Acceleration → SOC gradually decreases
- Braking → SOC increases due to negative current

The SOC increase during the braking window confirms:

- Energy recovery is working
- Battery charging power matches regen torque
- No unrealistic overcharging occurs (limit logic is correct)

The SOC recovery pattern is consistent with experimental plots from the reference study.



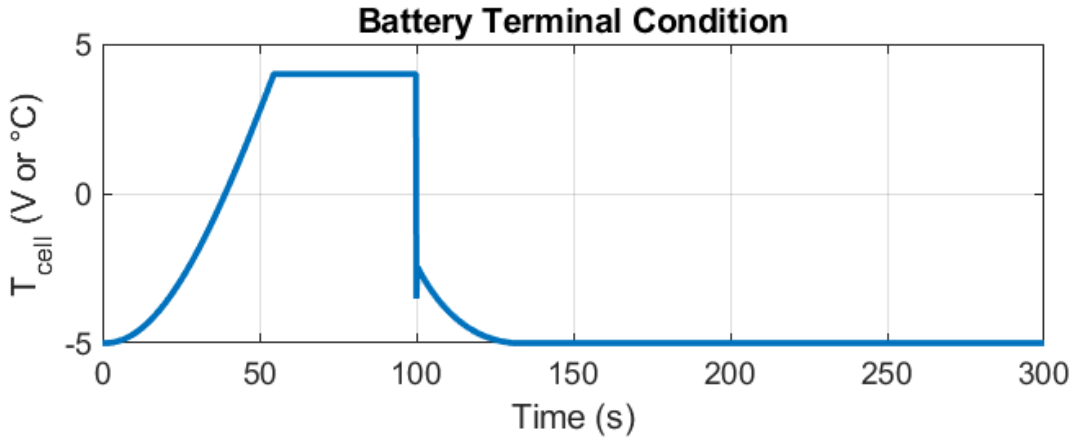
Graph 5: Battery SOC

3.6 Battery Terminal Voltage (T_{cell})

The T_{cell} signal remains within safe boundaries throughout the simulation(Graph 6). During braking:

- charging current increases terminal voltage
- lookup table prevents excessive charging
- voltage drop due to internal resistance is correctly modeled

This ensures safe and realistic operation of the regenerative braking system.



Graph 6: Battery Terminal Voltage (Tcell)

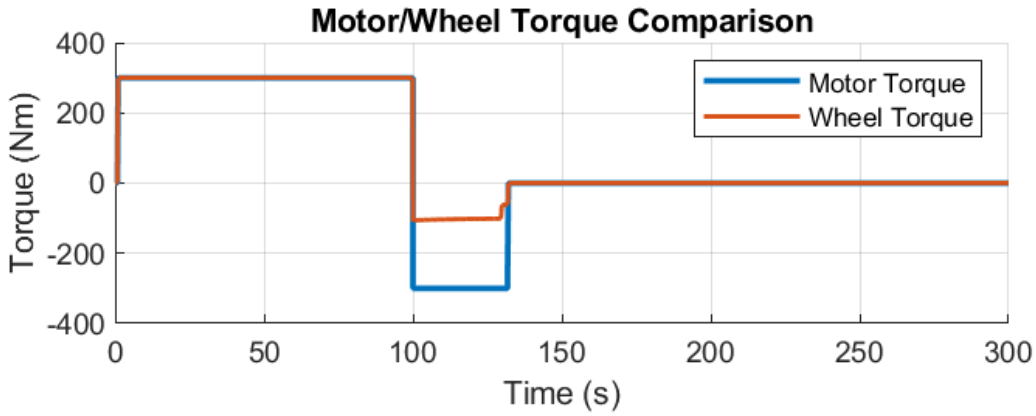
3.7 Torque Blending Behavior

The coordinated wheel torque is:

$$T_{wheel} = T_{req\_cmd} + T_{e\_cmd}$$

The blended torque curve demonstrates(Graph 7):

- seamless switch between traction and regen
- realistic reduction in regen effort as speed approaches zero
- stable behavior without oscillation



Graph 7: Torque Blending

4. Conclusions

In this project, a regenerative braking control strategy is modeled and analyzed using a MATLAB/Simulink electric vehicle powertrain framework. The proposed approach integrates battery State of Charge (SOC), motor speed, terminal cell temperature, and current limitations to generate a safe and efficient regenerative torque command. A 2-D lookup table based on battery voltage and motor speed is used to determine the maximum allowable regeneration power, while Stateflow logic ensures safe operation under different driving modes, such as acceleration, cruising, and braking.

Simulation results demonstrate that the system successfully applies negative motor torque during braking, recovers energy, and respects battery protection limits such as maximum charge current and thermal constraints. The SOC behavior, battery current profile, and motor/wheel torque comparison verify that the implemented controller behaves consistently with the regenerative braking characteristics described in the literature.

5. References

[1]A. Emadi, Y. J. Lee and K. Rajashekara, “Power Electronics and Motor Drives in Electric, Hybrid Electric, and Plug-In Hybrid Electric Vehicles,” IEEE Transactions on Industrial Electronics, vol. 55, no. 6, pp. 2237–2245, Jun. 2008.

[2]M. Ehsani, Y. Gao and A. Emadi, Modern Electric, Hybrid Electric, and Fuel Cell Vehicles: Fundamentals, Theory, and Design, 2nd ed. Boca Raton, FL, USA: CRC Press, 2010.

[3]H. Khaligh and Z. Li, “Battery, Ultracapacitor, Fuel Cell, and Hybrid Energy Storage Systems for Electric, Hybrid Electric, and Plug-In Hybrid Electric Vehicles: State of the Art,” IEEE Transactions on Vehicular Technology, vol. 59, no. 6, pp. 2806–2814, Jul. 2010.

[4]W. Gao, “Performance Comparison of a Battery-Powered Electric Vehicle and a Fuel Cell Electric Vehicle,” IEEE Transactions on Vehicular Technology, vol. 54, no. 3, pp. 836–846, May 2005.

[5]J. Cao and N. Schofield, “Regenerative Braking Control Strategies for Hybrid Electric Vehicles,” IEEE Transactions on Vehicular Technology, vol. 57, no. 6, pp. 3600–3611, Nov. 2008.

6. Appendix A – MATLAB Code for Regenerative Braking Simulation

```
clear; clc; close all;

model = 'Regenerative_Braking';

simTime = 300;   % Adjust according your simulation time

load_system(model);
```

========== Run Simulation ==========

```
simOut = sim(model, ...

   'StopTime', num2str(simTime), ...

   'SrcWorkspace', 'base', ...

   'SaveOutput', 'on');
```

========== Extract Logged Signals ==========

```
SOC      = simOut.get('SOC');          % From To Workspace block

Ibatt    = simOut.get('I_batt');

Tcell    = simOut.get('Tcell');

Speed    = simOut.get('speed');

Treq     = simOut.get('T_req_cmd');

Tm_req   = simOut.get('T_m_req');

WheelT   = simOut.get('Wheel_Torque_Nm');

Te_req   = simOut.get('T_e_cmd');

% ---------- Time Vector Generation ----------

% Each signal may have different sampling → generate time individually
```

```
t_SOC    = linspace(0, simTime, length(SOC));

t_Ibatt  = linspace(0, simTime, length(Ibatt));

t_Tcell  = linspace(0, simTime, length(Tcell));

t_spd    = linspace(0, simTime, length(Speed));

t_Treq   = linspace(0, simTime, length(Treq));

t_Tm     = linspace(0, simTime, length(Tm_req));

t_WT     = linspace(0, simTime, length(WheelT));

t_Te     = linspace(0, simTime, length(Te_req));
```

========== Plot Results ==========

```
figure('Name','Regenerative Braking Results','Color','w','Position',[150 80 1100 1200]);

% 1 – SOC

subplot(4,2,1);

plot(t_SOC, SOC, 'LineWidth', 1.8); grid on;

xlabel('Time (s)'); ylabel('SOC (%)'); title('Battery SOC');

% 2 – Battery Current

subplot(4,2,2);

plot(t_Ibatt, Ibatt, 'LineWidth', 1.8); grid on;

xlabel('Time (s)'); ylabel('I_{batt} (A)');

title('Battery Current');

% 3 – Cell Temperature / Terminal Voltage Signal

subplot(4,2,3);

plot(t_Tcell, Tcell, 'LineWidth', 1.8); grid on;
```

```
xlabel('Time (s)'); ylabel('T_{cell} (V or °C)');

title('Battery Terminal Condition');

% 4 – Vehicle Speed

subplot(4,2,4);

plot(t_spd, Speed, 'LineWidth', 1.8); grid on;

xlabel('Time (s)'); ylabel('Speed (km/h)');

title('Vehicle Speed');

% 5 – Requested Torque (ECU Output)

subplot(4,2,5);

plot(t_Treq, Treq, 'LineWidth', 1.8); grid on;

xlabel('Time (s)'); ylabel('T_{req} (Nm)');

title('Requested Torque from ECU');

% 6 – Motor/Mechanical Torque

subplot(4,2,6);

hold on; grid on;

plot(t_Tm, Tm_req, 'LineWidth', 1.8);

plot(t_WT, WheelT, 'LineWidth', 1.4);

xlabel('Time (s)'); ylabel('Torque (Nm)');

title('Motor/Wheel Torque Comparison');

legend('Motor Torque','Wheel Torque');

%7- Regenerative Torque

subplot(4,2,7);

hold on; grid on;

plot(t_Te, Te_req, 'LineWidth', 1.8);
```

```
xlabel('Time (s)'); ylabel('Torque (Nm)');

title('Regenerative Torque');
```