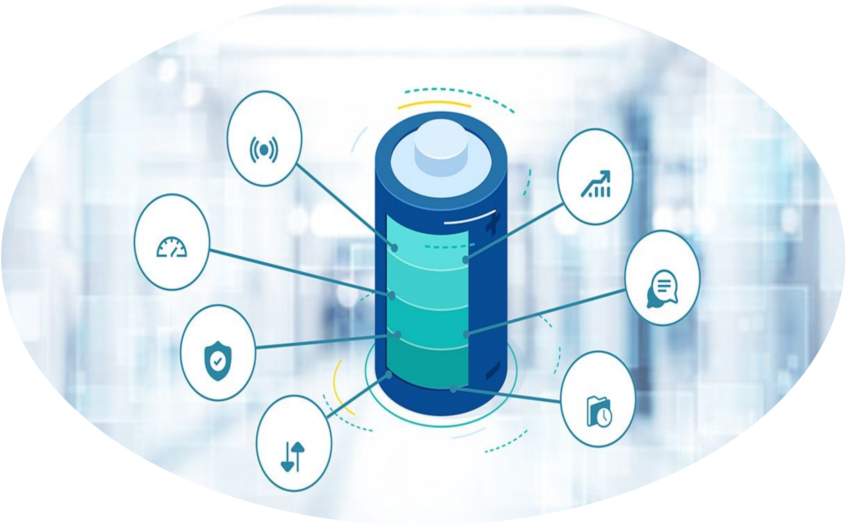
An introduction to Battery Management Systems



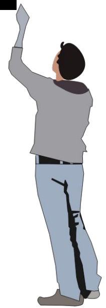
Dr. Adrian Guel-Cortez
PhD Physics | M.Eng. Electrical/Control Engineering | B.Eng. Mechatronics
Brill Power. Oxford, UK

GISR$e^{-s\tau}$

EL INGE DE CONTROL

A BMS is an embedded system

1. **Protects the safety of the battery operated device's operator.**
2. Detects unsafe operating conditions and responds.
3. Protects cells of battery from damage in abuse/failure cases.
4. Prolongs life of battery (normal operating cases).
5. Maintains battery in a state in which it can fulfill its functional design requirements.
6. Informs the application controller how to make the best use of the pack right now (e.g., power limits), control charger, etc.



Applications

- Vehicular applications include:
  - Hybrid-Electric Vehicle (HEV): Motive power provided by battery plus at least one other source (*e.g.*, gasoline engine). Essentially zero all-electric vehicle range.
  - Plug-in Hybrid-Electric Vehicle (PHEV): Larger battery than HEV allows some all-electric range under certain operating conditions.
  - Extended-Range Electric Vehicle (E-REV): Larger battery than PHEV allows some all-electric range under full-load conditions.
  - Electric Vehicle (EV), a.k.a. Battery-Electric Vehicle (BEV): Battery provides only motive power.

The world's largest electric vehicle (EV) companies

Tesla (USA)

- **Market Cap**: ~$600+ billion (varies)
- **Key Models**: Model S, Model 3, Model X, Model Y, Cybertruck, Semi, Roadster
- **Strengths**: Global leader in EV technology, battery efficiency, autonomous driving, and charging infrastructure.

BYD (China)

- **Market Cap**: ~$100+ billion
- **Key Models**: Han, Tang, Dolphin, Seal, Atto 3, Song
- **Strengths**: Strong domestic and international presence, backed by Warren Buffett's Berkshire Hathaway, leading in battery production.

Rivian (USA)

- **Market Cap**: ~$15-20 billion
- **Key Models**: R1T (electric truck), R1S (SUV), Amazon electric delivery vans
- **Strengths**: Strong backing from Amazon and Ford, focusing on adventure and commercial EVs.

Lucid Motors (USA)

- **Market Cap**: ~$10-15 billion
- **Key Models**: Lucid Air, Lucid Gravity (upcoming)
- **Strengths**: Luxury EVs with industry-leading range, backed by Saudi Arabia's Public Investment Fund (PIF).

NIO (China)

- **Market Cap**: ~$10-15 billion
- **Key Models**: ES8, ES6, EC6, ET7, ET5
- **Strengths**: Innovative battery swapping technology, focus on premium EVs.

Another large-scale application that justifies advanced battery management is for grid-storage and backup

GISRe$^{-s\tau}$ EL INGE DE CONTROL

Battery Energy Storage Systems (BESS) companies

Contemporary Amperex Technology Co., Limited (CATL)

- **Headquarters**: Ningde, China
- **Overview**: CATL is the world's largest battery manufacturer, specializing in lithium-ion batteries for electric vehicles and energy storage systems. In 2023, CATL led global battery deliveries for energy storage, holding over 40% market share.

Tesla, Inc.

- **Headquarters**: Palo Alto, California, USA
- **Overview**: Through its subsidiary, Tesla Energy, the company offers energy storage products like the Powerwall for residential use and the Megapack for large-scale applications. In 2023, Tesla deployed 14.7 gigawatt-hours of battery energy storage products, marking a 125% increase over the previous year.

BYD Company Limited

- **Headquarters**: Shenzhen, China
- **Overview**: Originally a battery manufacturer, BYD has expanded into electric vehicles and energy storage solutions. In 2023, BYD's energy storage battery deliveries reached 22 GWh, reflecting a 57% increase from the prior year.

LS Power

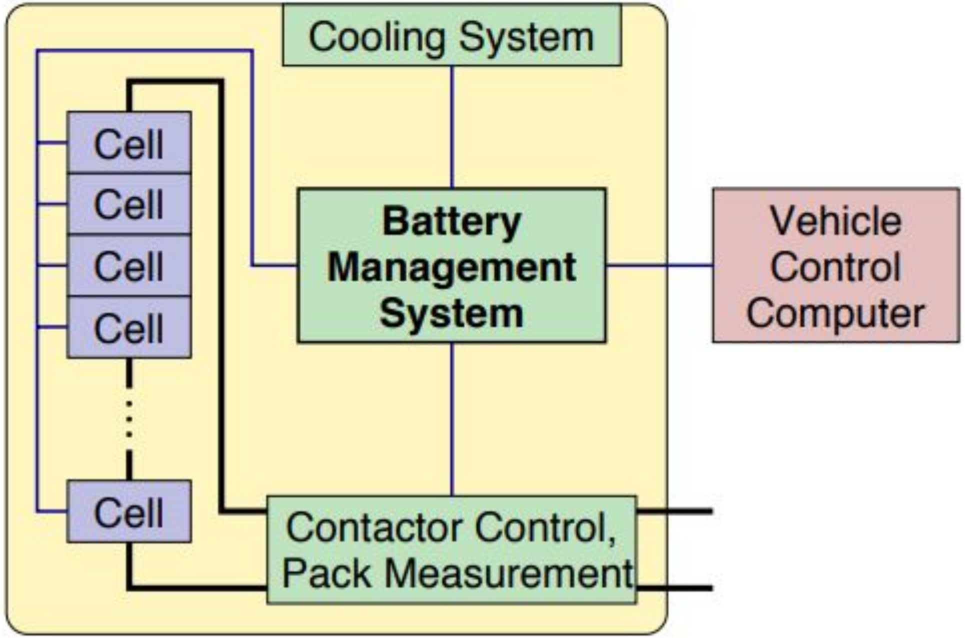
- **Headquarters**: New York, USA
- **Overview**: LS Power is an independent power company with a significant focus on energy storage. It operates the Gateway Energy Storage project in California, which, upon its commissioning in 2020, was the world's largest lithium-ion battery storage facility at 250 MW capacity.

Neoen

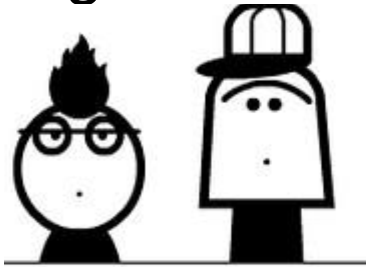
- **Headquarters**: Paris, France
- **Overview**: Neoen is a renewable energy company that develops and operates solar, wind, and energy storage projects. It operates the Victorian Big Battery in Australia, one of the world's most powerful batteries with a capacity of 300 MW/450 MWh.

BMS Functionality

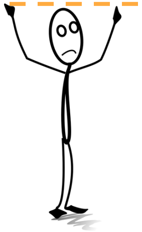
BMS is interconnected with all battery-pack components and with vehicle/grid control computer

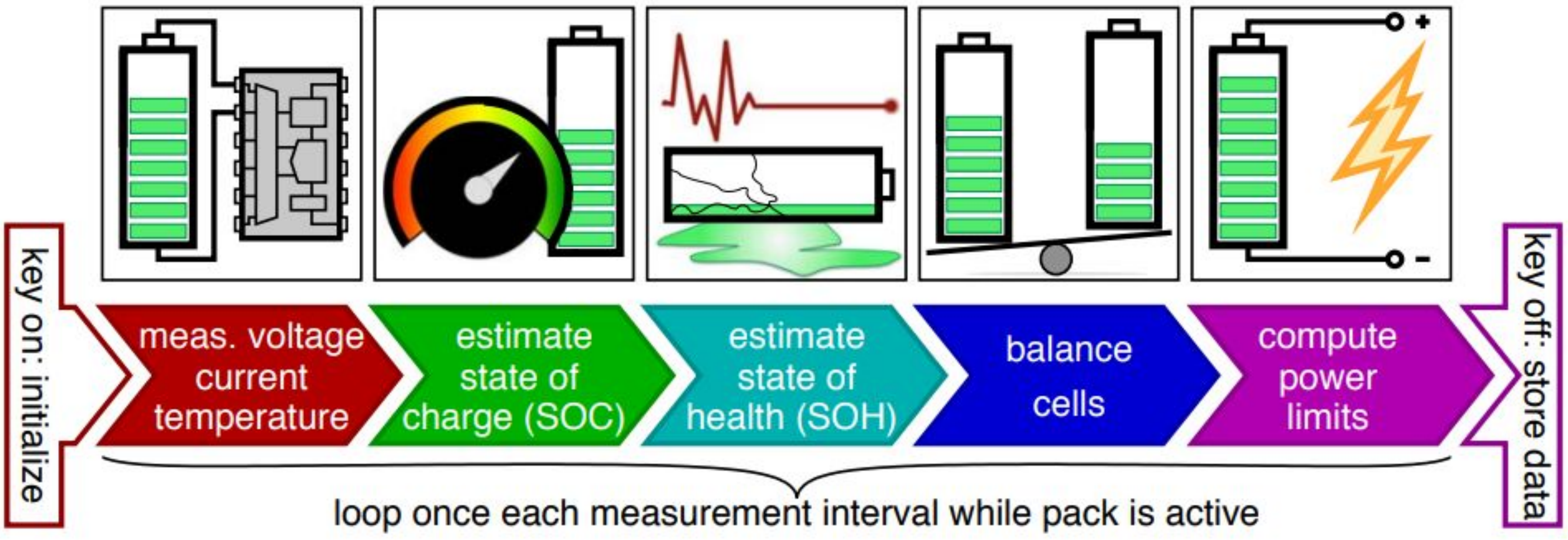


Functionality can be broken down into several categories:



1. **Sensing and high-voltage control:**
    a. Measure voltage
    b. Current
    c. Temperature
    d. Thermal management.
2. **Protection against:** Over-charge, over-discharge, over-current, short circuit, extreme temperatures.
3. **Interface:** Range estimation, communications, data recording, reporting.
4. **Performance management:** State-of-charge (SOC) estimation, power-limit computation, balance/equalize cells.
5. **Diagnostics:** Abuse detection, state-of-health (SOH) estimation, state-of-life (SOL) estimation.





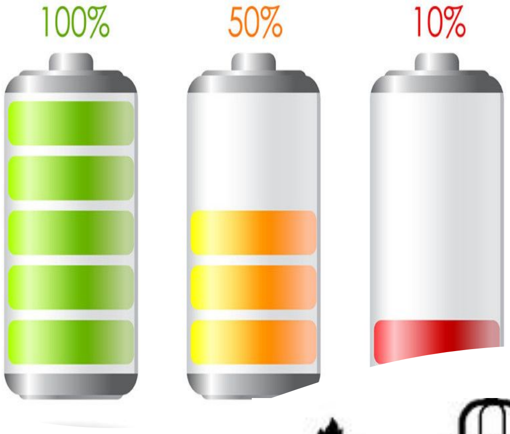
- Plett, G. L. (2015). *Battery management systems, Volume I: Battery modeling*. Artech House.
- Plett, G. L. (2015). *Battery management systems, Volume II: Equivalent-circuit methods*. Artech House.

Battery-pack sensing: Voltage

1. All cell voltages are measured in a lithium-ion pack:
   a. Indicator of relative balance of cells.
   b. Input to most SOC and SOH estimation algorithms.
2. It’s also a safety issue:
   a. Overcharging a lithium-ion cell can lead to “thermal runaway,” so we can’t skip measuring any voltages

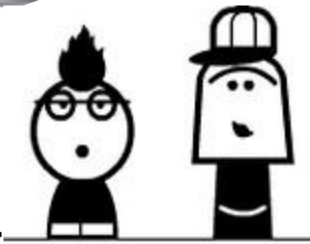
GISR$e^{-s\tau}$ EL INGE DE CONTROL

State-of-charge estimation

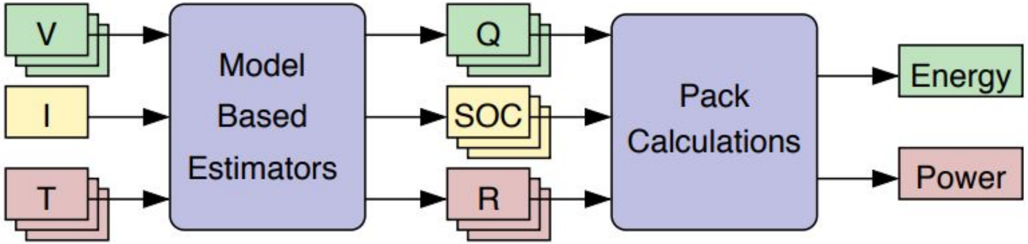


What needs to be estimated, and why?

1. xEVs need to know two battery quantities:
   a. How much energy is available in the battery pack;
   b. How much power is available in the immediate future.
2. An estimate of energy is most important for EV:
   a. Energy tells me how far I can drive.
3. An estimate of power is most important for HEV:
   a. Power tells me whether I can accelerate or accept braking charge.
4. Both are important for E-REV/PHEV

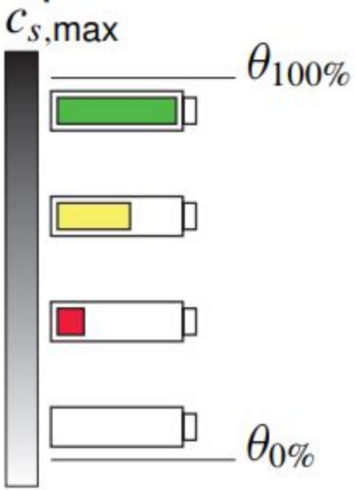


- To compute energy, we must know (at least) all cell states-of-charge and capacities.
- To compute power, we must know (at least) all cell states-of-charge and resistances.
- But, we cannot directly measure these parameters—we must estimate them as well.
- Available inputs include all **cell voltages, pack current, and temperatures of cells or modules.**



What really is state-of-charge (SOC)?

- Charging a cell moves lithium from the positive- to the negative-electrode of the cell; discharge does the opposite.
- Electrochemically, the cell state-of-charge (SOC) is related to average concentration of lithium in the negative-electrode solid particles.
- Define the present lithium concentration stoichiometry as $\theta = c_{s,\text{avg}}/c_{s,\text{max}}$.
- This stoichiometry is intended to remain between $\theta_{0\%}$ and $\theta_{100\%}$.
- Then, cell SOC is computed as: $z_k = (\theta - \theta_{0\%})/(\theta_{100\%} - \theta_{0\%})$.

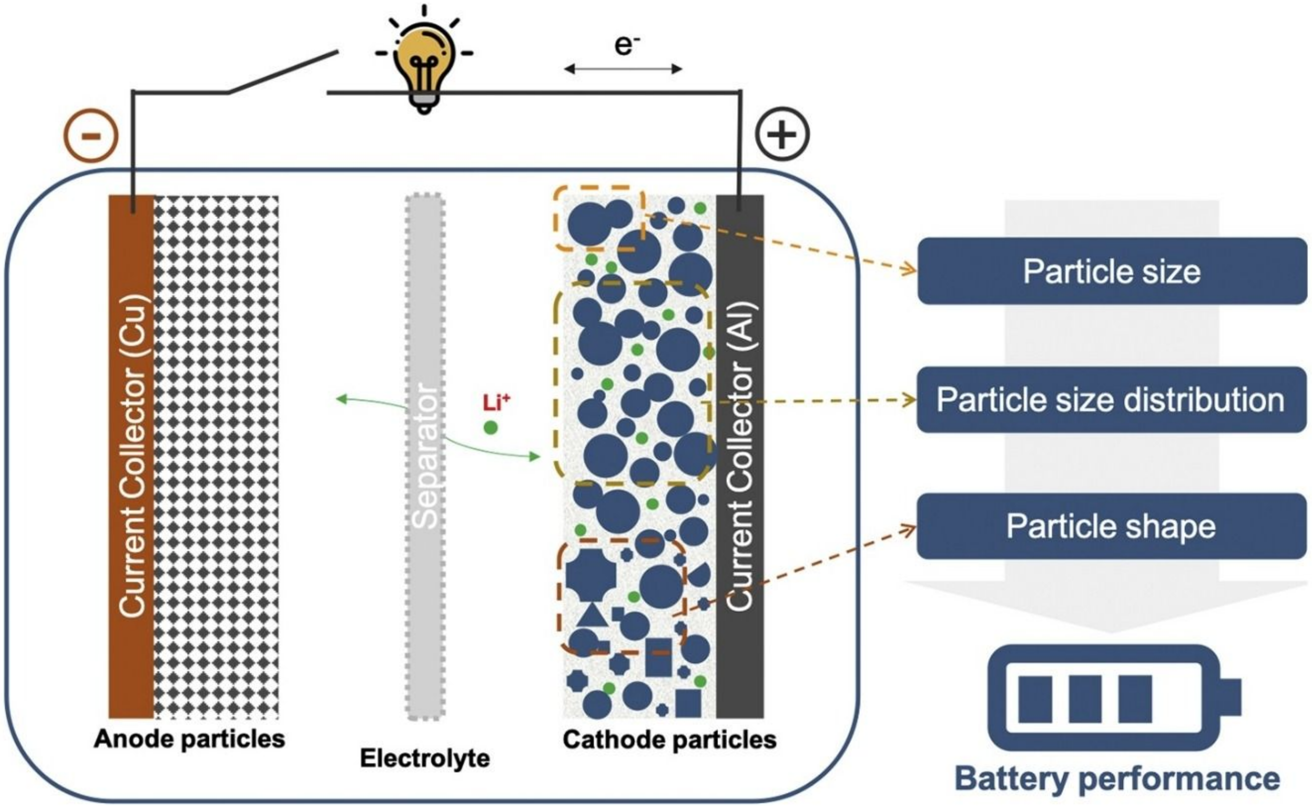


GISR$e^{-s\tau}$ EL INGE DE CONTROL

It is reasonable to wonder what is the coupling between SOC and cell voltage? Maybe I can infer SOC by measuring voltage?

- Cell voltage depends on temperature and electrode particle surface concentrations, but SOC depends on particle average concentrations.
  - Surface and average concentrations will not generally be the same.
- Furthermore,
  - Changing temperature changes cell voltage, but not average concentrations, so does not change SOC;
  - Resting a cell changes its voltage but not average concentrations, so does not change SOC;
  - History of cell usage changes steady-state surface concentration versus average concentration (hysteresis).

- In summary, SOC changes only due to passage of current, either charging or discharging the cell due to external circuitry, or due to self-discharge within the cell.
- So, we will find voltage useful as an indirect indicator of SOC, but not as a direct measurement of SOC.



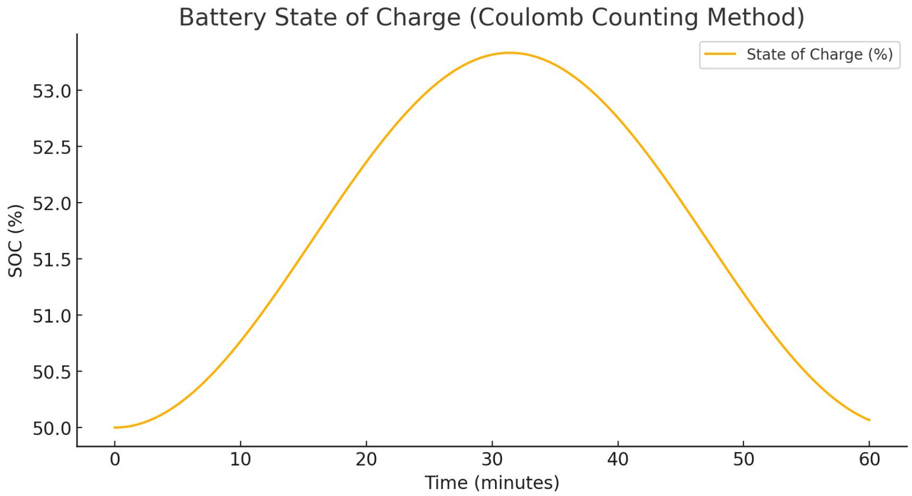
e-
Li+
Current Collector (Cu)
Separator
Current Collector (Al)
Anode particles
Electrolyte
Cathode particles
Particle size
Particle size distribution
Particle shape
Battery performance

coulomb counting

- How about current? SOC is related to cell current via
$$z(t) = z(0) - \frac{1}{Q}\int_0^t \eta i(\tau)\,\mathrm{d}\tau.$$
  - Cell current is positive on discharge, negative on charge.
  - η is cell coulombic efficiency ≈ 1 but ≤ 1.
  - Q is the cell total capacity in ampere seconds (coulombs).
- Note, total capacity Q is a measure of the number of locations in the electrode structure between $\theta_{0\%}$ and $\theta_{100\%}$ that could hold lithium.

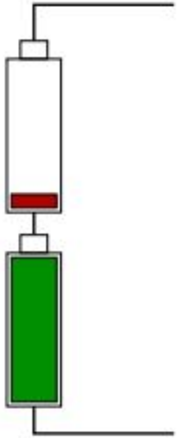
Example

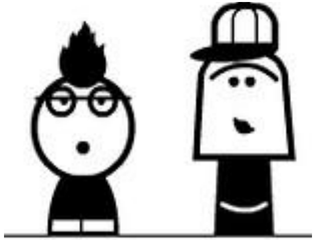
```
import numpy as np
import matplotlib.pyplot as plt

# Parameters
capacity_Ah = 50  # Battery capacity in Ah (Amp-hours)
initial_soc = 0.5  # Initial State of Charge (50%)
time_seconds = np.arange(0, 3600, 1)  # 1-hour simulation with 1s intervals
current_A = np.sin(time_seconds / 600) * 5  # Example varying current (sinusoidal)

# Convert capacity to Coulombs (1 Ah = 3600 Coulombs)
capacity_C = capacity_Ah * 3600
soc = np.zeros(len(time_seconds))
soc[0] = initial_soc

# Coulomb counting method to compute SOC
for i in range(1, len(time_seconds)):
    delta_charge = current_A[i]  # Current in Amperes (1s step = Coulombs)
    soc[i] = soc[i - 1] + delta_charge / capacity_C
    soc[i] = np.clip(soc[i], 0, 1)  # Ensure SOC stays between 0 and 1

# Plot results
plt.figure(figsize=(10, 5))
plt.plot(time_seconds / 60, soc * 100, label="State of Charge (%)")
plt.xlabel("Time (minutes)")
plt.ylabel("SOC (%)")
plt.title("Battery State of Charge (Coulomb Counting Method)")
plt.legend()
plt.grid()
plt.show()
```



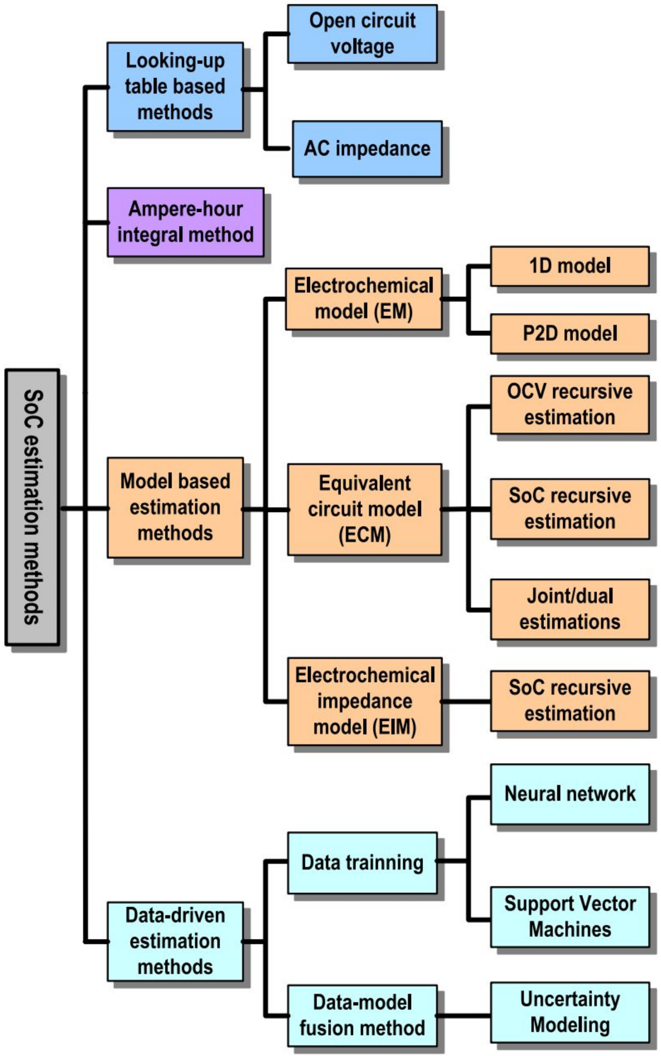
GISRe^{-sτ} EL INGE DE CONTROL

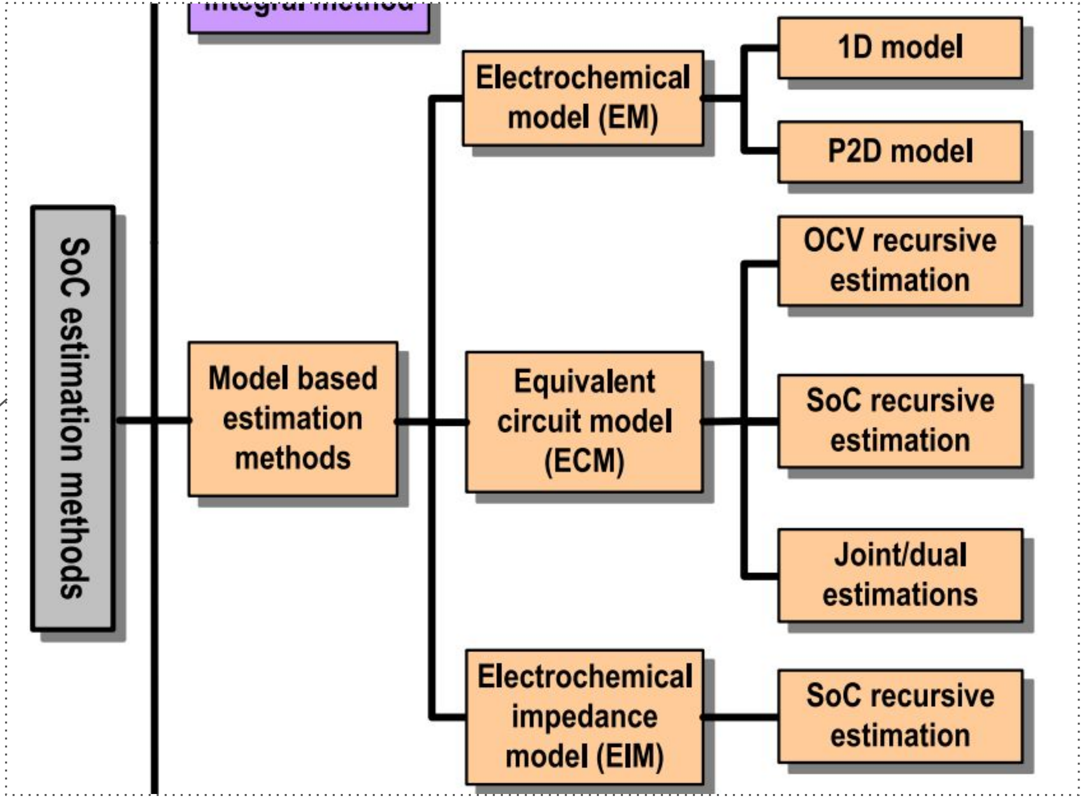
- Consider the picture to the right. What is the pack SOC?
  - Should it be 0 % because we cannot discharge?
  - Should it be 100 % because we cannot charge?
  - Should it be the average of the two, 50 %?
- The term “pack SOC” is ill-defined, and should not be used.





GISR$e^{-s\tau}$ EL INGE DE CONTROL



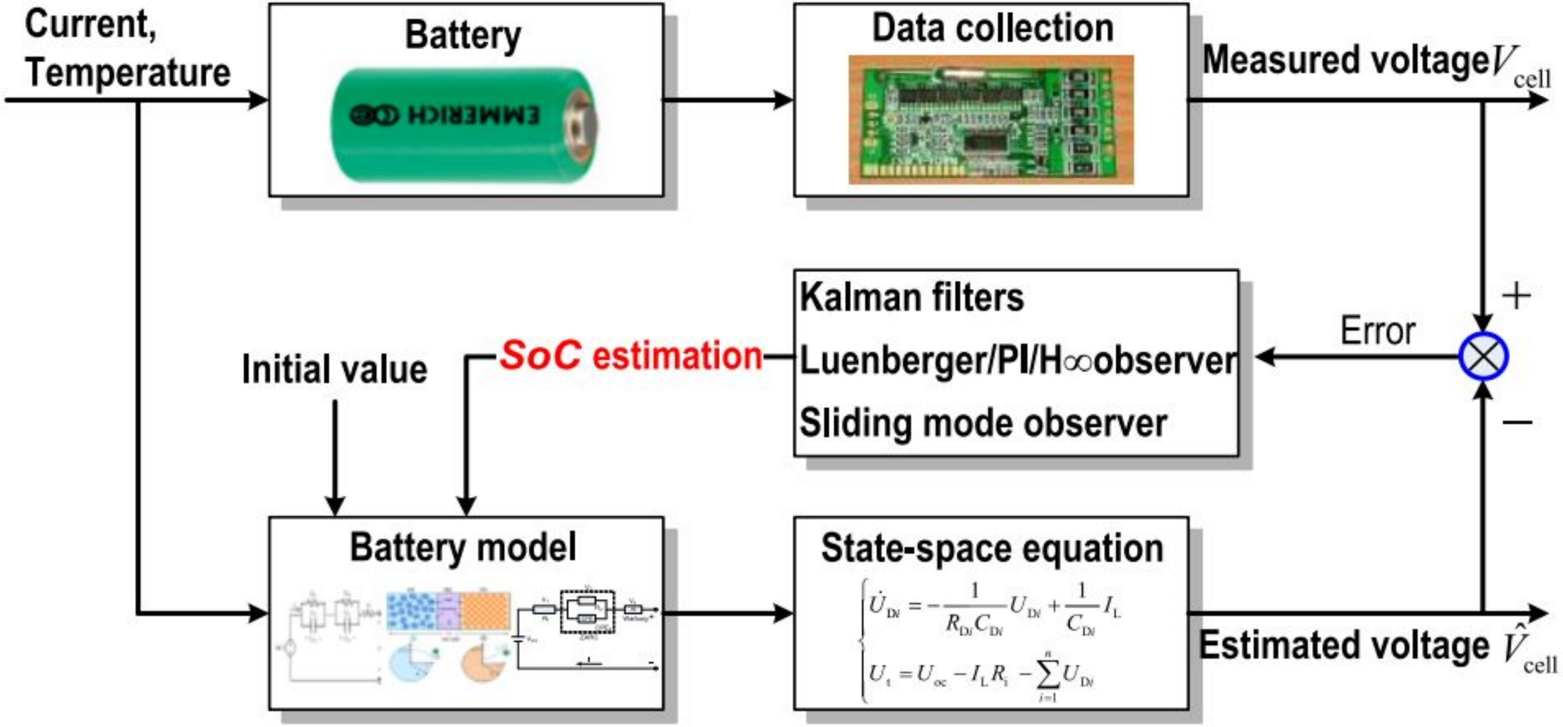


How, D. N., Hannan, M. A., Lipu, M. H., & Ker, P. J. (2019). State of charge estimation for lithium-ion batteries using model-based and data-driven methods: A review. *Ieee Access*, *7*, 136116-136136.

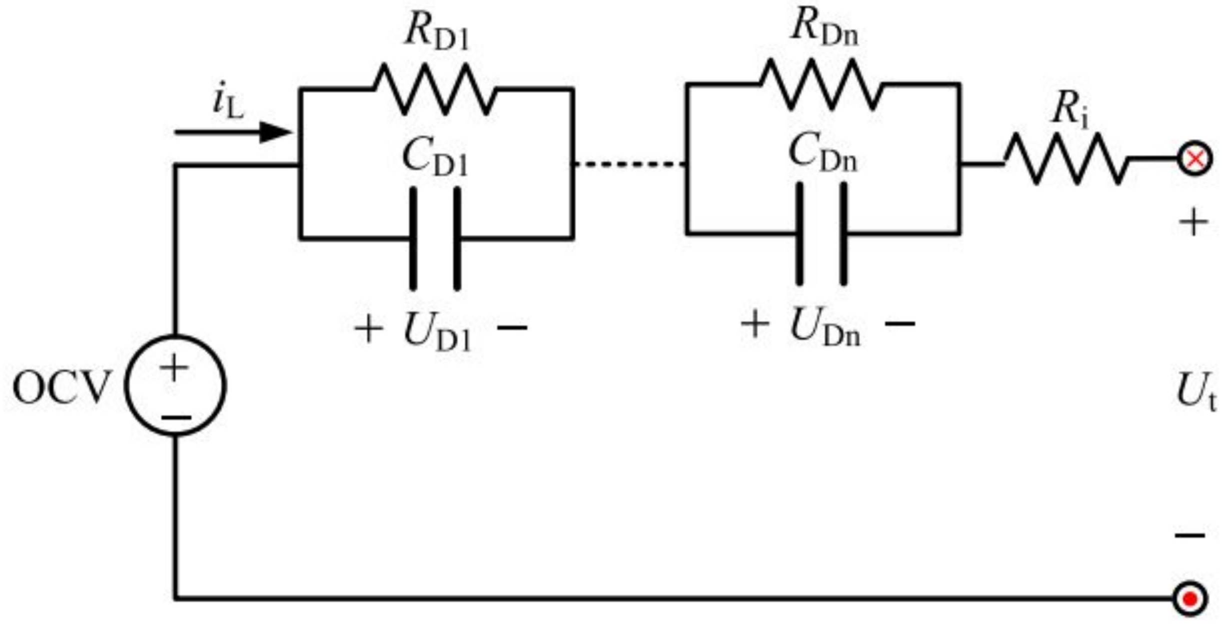
uel-Cortez

GISR$e^{-s\tau}$

EL INGE DE CONTROL



How, D. N., Hannan, M. A., Lipu, M. H., & Ker, P. J. (2019). State of charge estimation for lithium-ion batteries using model-based and data-driven methods: A review. *Ieee Access*, *7*, 136116-136136.

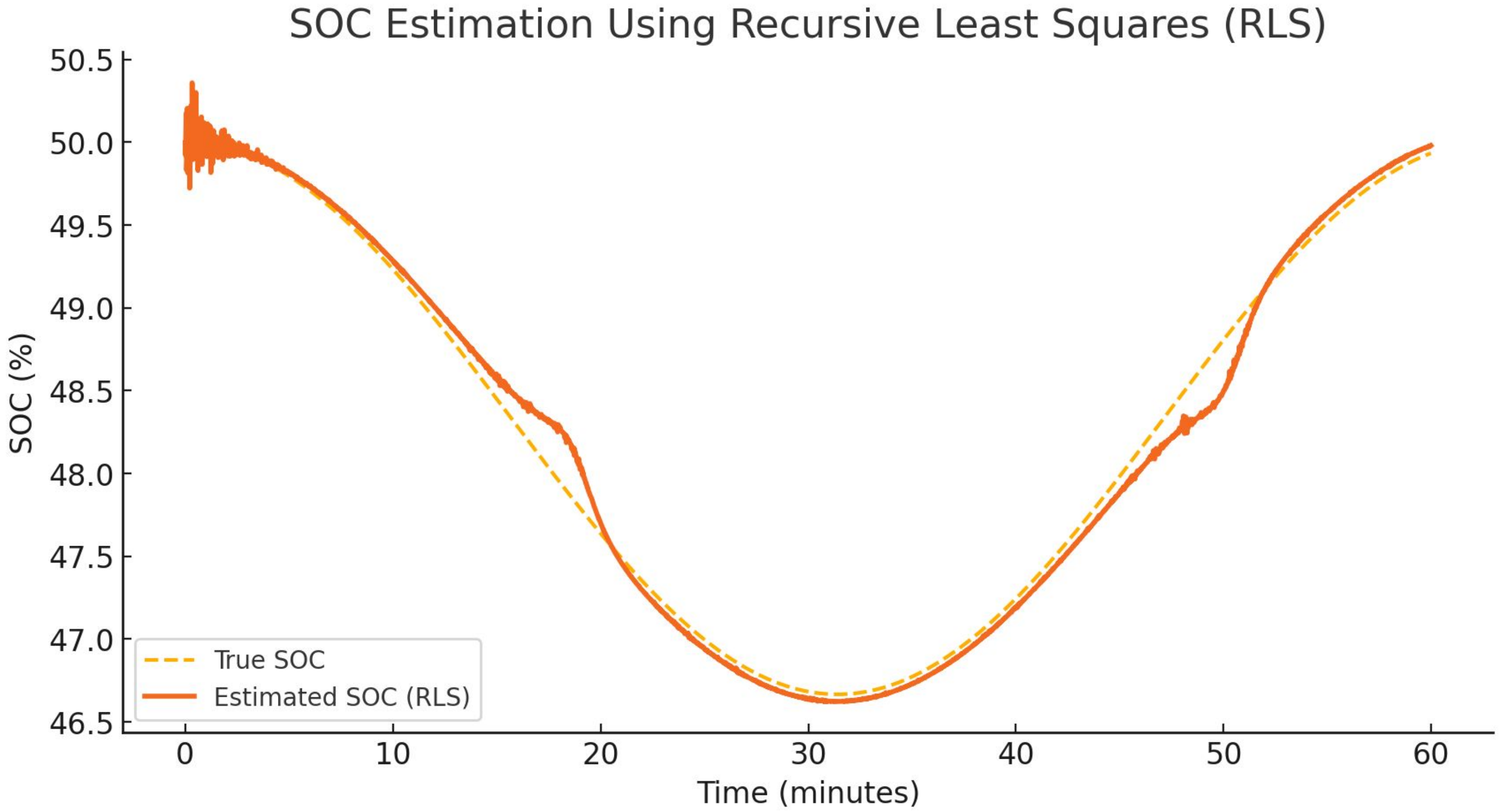


How, D. N., Hannan, M. A., Lipu, M. H., & Ker, P. J. (2019). State of charge estimation for lithium-ion batteries using model-based and data-driven methods: A review. *Ieee Access*, *7*, 136116-136136.

SoC estimation example

```python
import numpy as np
import matplotlib.pyplot as plt

# Simulation parameters
np.random.seed(42)
time_seconds = np.arange(0, 3600, 1)  # 1-hour simulation with 1s intervals

# Battery parameters (simplified model)
capacity_Ah = 50  # Battery capacity in Ah
capacity_C = capacity_Ah * 3600  # Convert to Coulombs
initial_soc = 0.5  # Initial SOC (50%)
internal_resistance = 0.01  # Internal resistance in ohms
open_circuit_voltage = 3.7 + 0.5 * (initial_soc - 0.5)  # Simplified OCV model

# Generate synthetic current profile (varying sinusoidal current)
current_A = np.sin(time_seconds / 600) * 5

# Compute true SOC using Coulomb counting
true_soc = np.zeros(len(time_seconds))
true_soc[0] = initial_soc

for i in range(1, len(time_seconds)):
    delta_charge = current_A[i]  # Current in Coulombs per second
    true_soc[i] = true_soc[i - 1] - delta_charge / capacity_C
    true_soc[i] = np.clip(true_soc[i], 0, 1)  # Keep SOC in bounds

# Generate synthetic voltage measurements using a simplified model
measured_voltage = open_circuit_voltage - internal_resistance * current_A
measured_voltage += np.random.normal(0, 0.02, len(measured_voltage))  # Add
noise

# Recursive Least Squares (RLS) initialization
lam = 0.99  # Forgetting factor
P = np.eye(3) * 1000  # Large initial covariance matrix
theta = np.zeros(3)  # Initial parameter estimates
estimated_soc = np.zeros(len(time_seconds))

# Recursive Least Squares Algorithm
for i in range(len(time_seconds)):
    phi = np.array([measured_voltage[i], current_A[i], 1])  # Regression vector
    K = P @ phi / (lam + phi.T @ P @ phi)  # Gain computation
    error = true_soc[i] - phi.T @ theta  # Compute error
    theta += K * error  # Update parameter estimate
    P = (P - np.outer(K, phi.T @ P)) / lam  # Update covariance matrix

    # Estimated SOC using updated parameters
    estimated_soc[i] = theta[0] * measured_voltage[i] + theta[1] * current_A[i] + theta[2]

# Ensure SOC remains between 0 and 1
estimated_soc = np.clip(estimated_soc, 0, 1)

# Plot results
plt.figure(figsize=(10, 5))
plt.plot(time_seconds / 60, true_soc * 100, label="True SOC", linestyle="dashed")
plt.plot(time_seconds / 60, estimated_soc * 100, label="Estimated SOC (RLS)",
linewidth=2)
plt.xlabel("Time (minutes)")
plt.ylabel("SOC (%)")
plt.title("SOC Estimation Using Recursive Least Squares (RLS)")
plt.legend()
plt.grid()
plt.show()
```

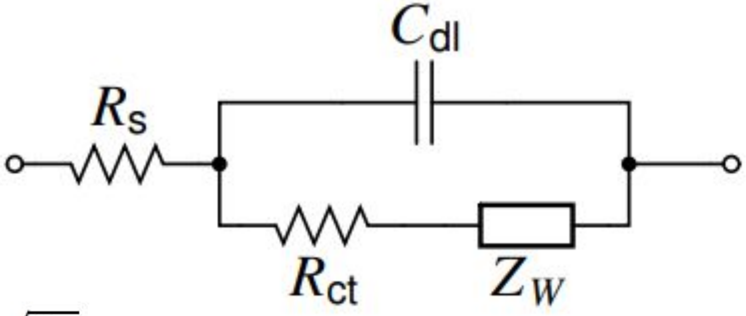

GISR$e^{-s\tau}$ EL INGE DE CONTROL



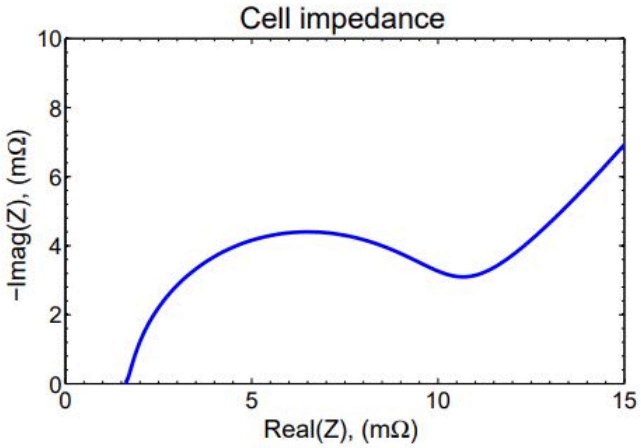
GISR$e^{-s\tau}$

EL INGE DE CONTROL

Interesting modelling methods



- The frequency-dependent Warburg impedance $Z_W = A_W/\sqrt{j\omega}$ models diffusion of lithium ions in the electrodes.
- The phase contributed to the circuit by this element is $45°$, which is most easily observed in a Nyquist plot of a cell's electrochemical impedance spectrum as a straight line at $45°$ at low frequency.



Fractional calculus!!!

Interesting cell-balancing algorithms

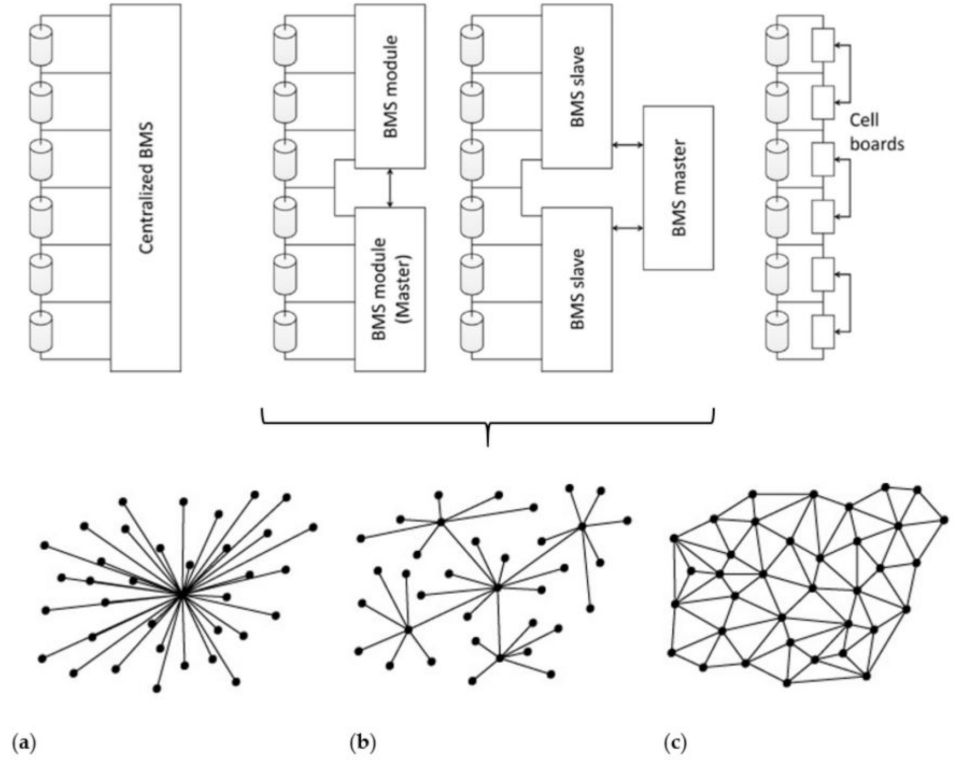


Figure 4. Basic types of BMS topologies (**top**) and their corresponding control networks (**bottom**): (**a**) centralized; (**b**) decentralized (modular and master-slave topologies), and (**c**) distributed.

Barreras, J. V., de Castro, R., Wan, Y., & Dragicevic, T. (2021). A consensus algorithm for multi-objective battery balancing. *Energies*, *14*(14), 4279.

Itagi, A. R., Kallimani, R., Pai, K., Iyer, S., López, O. L., & Mutagekar, S. (2024). Cell Balancing Paradigms: Advanced Types, Algorithms, and Optimization Frameworks. *arXiv preprint arXiv:2411.05478*.

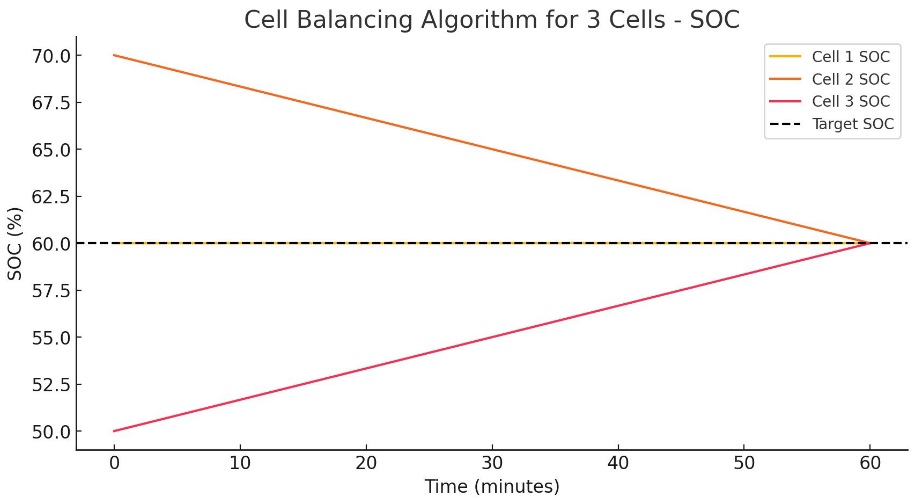
GISRe$^{-s\tau}$

EL INGE DE CONTROL

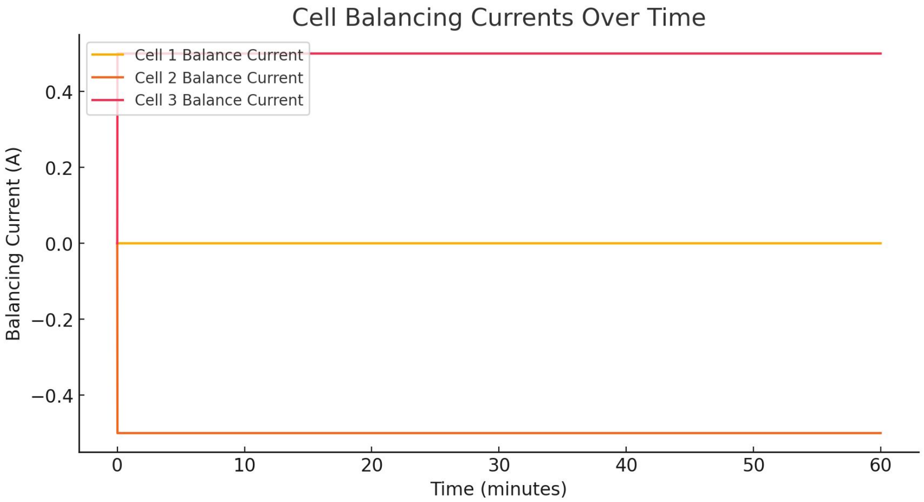
Example dummy cell-balancing algorithm

```
import numpy as np
import matplotlib.pyplot as plt

# Simulation parameters
time_seconds = np.arange(0, 3600, 1)  # 1-hour simulation with 1s
intervals
dt = 1  # Time step in seconds

# Battery parameters
capacity_Ah = 5  # Each cell capacity in Ah
capacity_C = capacity_Ah * 3600  # Convert to Coulombs
initial_soc = np.array([0.6, 0.7, 0.5])  # Initial SOC for the 3 cells
internal_resistance = 0.01  # Internal resistance in ohms

# Balancing control parameters
balance_current_max = 0.5  # Maximum balancing current (A)
soc_target = np.mean(initial_soc)  # Target SOC is the average of all
cells

# Arrays to store SOC values over time
soc_history = np.zeros((len(time_seconds), 3))
soc_history[0] = initial_soc
```

```
# Control loop for cell balancing
for t in range(1, len(time_seconds)):
    soc_current = soc_history[t - 1]

    # Compute control currents for balancing
    balance_current = np.zeros(3)
    for i in range(3):
        if soc_current[i] > soc_target:
            balance_current[i] = -balance_current_max  # Discharge extra energy
        elif soc_current[i] < soc_target:
            balance_current[i] = balance_current_max  # Charge to balance

    # Update SOC values based on balancing currents
    new_soc = soc_current + (balance_current / capacity_C) * dt
    soc_history[t] = np.clip(new_soc, 0, 1)  # Keep SOC within limits

# Plot results
plt.figure(figsize=(10, 5))
plt.plot(time_seconds / 60, soc_history[:, 0] * 100, label="Cell 1 SOC")
plt.plot(time_seconds / 60, soc_history[:, 1] * 100, label="Cell 2 SOC")
plt.plot(time_seconds / 60, soc_history[:, 2] * 100, label="Cell 3 SOC")
plt.axhline(y=soc_target * 100, color='k', linestyle='dashed', label="Target SOC")
plt.xlabel("Time (minutes)")
plt.ylabel("SOC (%)")
plt.title("Cell Balancing Algorithm for 3 Cells")
plt.legend()
plt.grid()
plt.show()
```

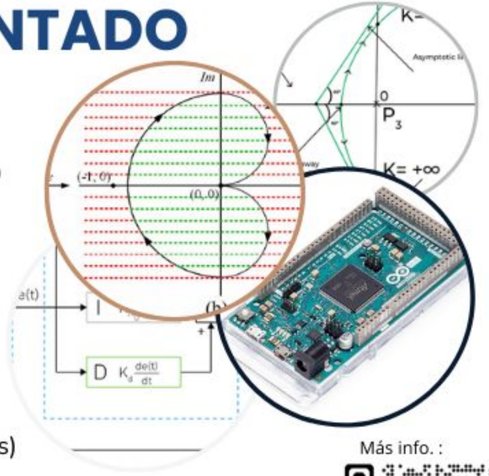


Cell Balancing Algorithm for 3 Cells - SOC
Cell 1 SOC
Cell 2 SOC
Cell 3 SOC
Target SOC
SOC (%)
70.0
67.5
65.0
62.5
60.0
57.5
55.0
52.5
50.0
0
10
20
30
40
50
60
Time (minutes)



Cell Balancing Currents Over Time
Cell 1 Balance Current
Cell 2 Balance Current
Cell 3 Balance Current
Balancing Current (A)
0.4
0.2
0.0
−0.2
−0.4
0
10
20
30
40
50
60
Time (minutes)

TALLER DE SISTEMAS DE CONTROL RETROALIMENTADO



Temario:

1. Introducción al Control de Retroalimentación (2 hr)
2. Modelado de Sistemas en el Dominio del Tiempo (2 hrs)
3. Respuesta Dinámica (2 hrs)
4. Propiedades Básicas de la Retroalimentación (2 hr)
5. Análisis de Estabilidad (4 hrs)
6. Análisis del Lugar de las Raíces (2 hrs)
7. Diseño de Controladores del Lugar de las Raíces (2 hrs)
8. Análisis de Respuesta en Frecuencia (2 hrs)
9. Diseño de Respuesta en Frecuencia (2 hrs)
10. Implementación de Controlador Digital (Arduino) (2 hrs)

Más info. :

Sábados 11:00 - 13:00 hrs y 15:00 - 17:00 hrs tiempo de la ciudad de México.

Del 22 de Febrero al 29 de Marzo.

120 USD

INSTRUCTORES:

Dr Enrique Diez
Sistemas con retardo de tiempo, electrónica de potencia, variable compleja

Dr. Bryan Rojas R.
Robótica, visión artificial, sistemas con retardo de tiempo, aplicaciones industriales.

Mtro. Juan José Meza G.
Algoritmos de estimación, visión por computadora, robótica

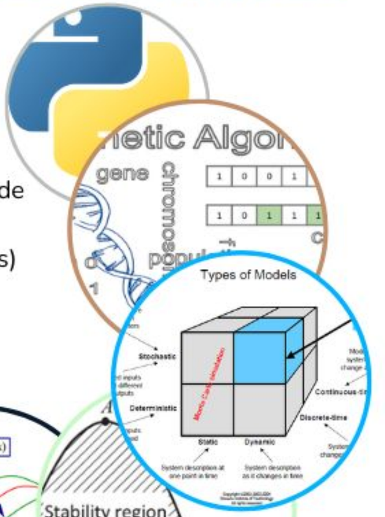
EL INGE DE CONTROL

@elingedecontrol

elingedecontrol

Participa!!

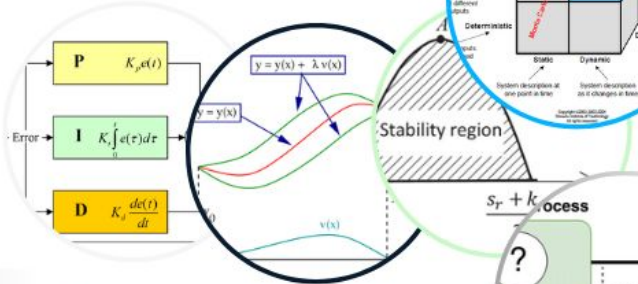
TALLER TEMAS SELECTOS DE INGENIERÍA DE CONTROL I

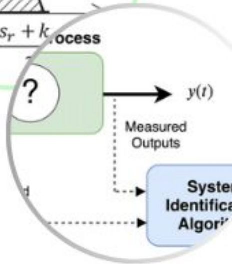


Temario:

- Modelado matemático y simulación de sistemas dinámicos (4hrs)
- Programación de controladores PID y comparación con otro tipo de controladores de bajo orden (2hrs)
- Ejemplo: Diseño y control de un sistema carro con péndulo invertido (2hrs)
- Diseño de controladores PID por algorítmos Evolutivos (2hrs)
- Diseño de controladores PID por el método D-composition (2 hrs)
- Identificación paramétrica de sistemas lineales y no lineales (3 hrs)

Más info. :





INSTRUCTOR:

Dr. Adrian Guel-Cortez
Modelado matemático,
teoría de estimación,
sistemas estocásticos

Martes y Jueves 12:00 - 13:00 hrs tiempo de la ciudad de México.

18 de Febrero

80 USD

EL INGE DE CONTROL

@elingedecontrol

elingedecontrol

Participa!!

Thanks

Email: adrianjguelc@gmail.com

Facebook: facebook.com/elingedecontrol/

Youtube: youtube.com/@elingedecontrol


1:07:40
Explorando el Cálculo Fraccionario y sus Aplicaciones Interdisciplinarias
134 views • Streamed 8 days ago
1:20:35
La Esencia y Trascendencia de la Publicación Científica
58 views • Streamed 8 days ago
1:03:11
Diseño automático de comportamientos colectivos para...
93 views • Streamed 2 weeks ago
1:47:31
Investigación en Sistemas con Retardo (GISRe)
210 views • Streamed 3 weeks ago
1:12:41
Del Laboratorio al mundo automotriz ft Christian Trejo
381 views • Streamed 4 weeks ago
1:02:10
Innovación en prótesis robóticas: Superando desafíos científicos y n...
79 views • Streamed 1 month ago
1:28:25
la cibernética en sociedad
95 views • Streamed 1 month ago
1:30:51
Vivir la Ingenieria de Control
236 views • Streamed 1 month ago