LAB 08 | المختبر 08

Integrated PV-Battery Smart Energy Management

الإدارة الذكية المتكاملة لنظام PV والبطارية

PV Generation, Load, Battery Storage, Grid Exchange, and Electricity Price

توليد PV والحمل وتخزين الطاقة والتبادل مع الشبكة وأسعار الكهرباء

Prepared by | إعداد: Dr.-Ing. Aouss Gabash

🎯 Lab Objectives

- Integrate PV generation, electrical load, battery storage, and the utility grid.
- Formulate the power-balance equation.
- Model charging and discharging of the battery.
- Calculate grid import and export.
- Use electricity prices in an energy-management objective.
- Optimize the battery schedule.
- Evaluate PV self-consumption and grid-energy reduction.

🎯 أهداف المختبر

- دمج توليد PV والحمل والبطارية والشبكة الكهربائية في نموذج واحد.
- صياغة معادلة توازن القدرة.
- نمذجة شحن وتفريغ البطارية.
- حساب الاستيراد والتصدير من الشبكة.
- إدخال أسعار الكهرباء ضمن دالة إدارة الطاقة.
- تحسين جدول تشغيل البطارية.

- تقييم الاستهلاك الذاتي للطاقة الشمسية وتقليل الطاقة المسحوبة من الشبكة.

1. Smart Energy System

The energy-management system contains four main components:

PV System → Load ↔ Battery ↔ Grid

The energy manager decides when the battery should charge or discharge.

1. نظام إدارة الطاقة الذكي

يتكون النظام من توليد شمسي وحمل وبطارية وشبكة كهربائية.

مهمة نظام إدارة الطاقة هي تنسيق تدفق القدرة بين هذه العناصر بطريقة اقتصادية وآمنة.

2. Power-Balance Equation

Using the battery sign convention:

$$\mathrm{PB} > 0 \longrightarrow \text{Battery Discharge}$$

$$\mathrm{PB} < 0 \longrightarrow \text{Battery Charge}$$

The grid power is:

$$Pgrid(k) = Pload(k) - PPV(k) - PB(k)$$

2. معادلة توازن القدرة

$$Pgrid(k) = Pload(k) - PPV(k) - PB(k)$$

إذا كان توليد PV والبطارية غير كافيين لتغذية الحمل، يتم شراء الفرق من الشبكة.

3. Grid Import and Export

$$\mathrm{Pgrid} > 0 \longrightarrow \text{Import}$$

$$\mathrm{Pgrid} < 0 \longrightarrow \text{Export}$$

A negative grid-power value means excess local energy is exported.

3. الاستيراد والتصدير من الشبكة

- Pgrid موجبة → شراء طاقة من الشبكة.
- Pgrid سالبة → تصدير فائض إلى الشبكة.

4. Example Profiles

Hour	Load [MW]	PV [MW]	Buy Price [€/MWh]
1	5.0	0.0	45
2	4.8	0.5	40
3	4.5	2.0	35
4	5.0	4.5	45
5	6.5	3.0	85
6	7.0	0.5	95

4. بيانات النظام

لدينا حمل وتوليد شمسي وسعر شراء مختلف لكل ساعة.

يظهر توليد PV في ساعات النهار، بينما يرتفع السعر في الساعات الأخيرة.

5. Battery Parameters

Parameter	Value
Energy Capacity	5 MWh
Initial SOC	50%
Minimum SOC	20%
Maximum SOC	90%
Maximum Charge Power	2 MW
Maximum Discharge Power	2 MW
Charge Efficiency	95%
Discharge Efficiency	95%

5. معاملات البطارية

نستخدم بطارية بسعة 5 MWh وحدود SOC من 20% إلى 90%.

6. Energy-Management Objective

The simplest objective is to minimize electricity-purchase cost:

$$J = \sum Cgrid(k)$$

where:

$$Cgrid(k) = Pricebuy(k)Pimport(k)\Delta t$$

6. دالة هدف إدارة الطاقة

نبدأ بهدف بسيط هو تقليل تكلفة شراء الطاقة من الشبكة.

$$J = \sum Pricebuy(k)Pimport(k)\Delta t$$

7. Import Power

Only positive grid power corresponds to purchased electricity:

$$Pimport(k) = \max(Pgrid(k), 0)$$

7. قدرة الاستيراد

$$Pimport(k) = \max(Pgrid(k), 0)$$

إذا كان Pgrid سالبًا فلا توجد طاقة مشتراة في تلك الساعة.

8. Optional Export Revenue

If exported electricity is compensated:

$$Pexport(k) = \max(-Pgrid(k), 0)$$

Then:

$$Revenue = \sum Pricesell(k) Pexport(k) \Delta t$$

and the net objective becomes:

$$J = PurchaseCost - ExportRevenue$$

8. عائد تصدير الطاقة

إذا كانت الشبكة تسمح ببيع الطاقة الفائضة، يمكن إضافة عائد التصدير إلى المسألة.

$$J = \text{تكلفة الشراء} - \text{عائد التصدير}$$

9. Battery Energy Model

Charging:

$$E(k+1) = E(k) + \eta ch|PB(k)|\Delta t$$

Discharging:

$$E(k+1) = E(k) - PB(k)\Delta t/\eta dis$$

9. نموذج طاقة البطارية

تتغير الطاقة المخزنة حسب الشحن أو التفريغ مع أخذ الكفاءة في الحسبان.

10. SOC Constraints

$$SOCmin \leq SOC(k) \leq SOCmax$$

For this case:

$$0.20 \leq SOC(k) \leq 0.90$$

10. قيود SOC

يجب الحفاظ على البطارية ضمن المجال التشغيلي الآمن طوال الفترة.

11. Decision Vector

For six hours:

$$x = [PB(1), PB(2), \dots, PB(6)]$$

Each candidate solution is therefore a complete battery schedule.

11. متجه القرار

$$x = [PB(1), PB(2), \dots, PB(6)]$$

يمثل كل كروموسوم جدول تشغيل كامل للبطارية.

12. Baseline Without Battery

Before optimization:

$$PB(k) = 0$$

therefore:

$$Pgrid, base = Pload - PPV$$

This provides a reference cost for comparison.

12. الحالة المرجعية دون البطارية

نحسب أولًا تكلفة النظام عندما لا تستخدم البطارية إطلاقًا.

يجب دائمًا مقارنة الحل المحسن بحالة مرجعية واضحة.

13. MATLAB Input Data

```
loadMW = [5.0 4.8 4.5 5.0 6.5 7.0];

pvMW = [0.0 0.5 2.0 4.5 3.0 0.5];

buyPrice = [45 40 35 45 85 95];

N = length(loadMW);

dt = 1;
```

13. البيانات في MATLAB

نخزن الحمل وقدرة PV وسعر شراء الكهرباء في متجهات زمنية.

14. Baseline Calculation

```
PgridBase = loadMW - pvMW;

PimportBase = max(PgridBase,0);

baseCost = sum(buyPrice .* PimportBase * dt);
```

14. حساب الحالة المرجعية

نحسب الاستيراد من الشبكة والتكلفة من دون استخدام البطارية.

15. SOC Penalty

If the battery violates its limits:

$$Jpen = J + PenaltySOC$$

A large penalty coefficient ensures infeasible schedules are unattractive.

15. عقوبة SOC

أي جدول تشغيل يؤدي إلى تجاوز حدود حالة الشحن يحصل على عقوبة كبيرة.

16. Final SOC Condition

We also require:

$$SOCfinal \approx SOCinitial$$

This prevents the optimizer from artificially lowering cost simply by emptying the battery at the end.

16. شرط SOC النهائي

نطلب عودة البطارية تقريبًا إلى حالة الشحن الابتدائية عند نهاية أفق الجدولة.

17. Complete MATLAB Program

```
clear; clc; close all;

% ------------------------------------------------
% Integrated PV-Battery Smart Energy Management
% Educational evolutionary optimization example
% ------------------------------------------------

loadMW = [5.0 4.8 4.5 5.0 6.5 7.0];

pvMW = [0.0 0.5 2.0 4.5 3.0 0.5];

buyPrice = [45 40 35 45 85 95];

N = length(loadMW);

dt = 1;

% ------------------------------------------------
% Battery parameters
% ------------------------------------------------

Emax = 5;

SOC0 = 0.50;

SOCmin = 0.20;
SOCmax = 0.90;

PchMax = 2;
PdisMax = 2;

etaCh = 0.95;
etaDis = 0.95;

% ------------------------------------------------
% Optimization parameters
% ------------------------------------------------

popSize = 70;
```

```
maxGen = 180;

Pc = 0.8;
Pm = 0.12;

sigmaMut = 0.4;

lambdaSOC = 1e5;
lambdaFinal = 1e5;

% ------------------------------------------------
% Baseline without battery
% ------------------------------------------------

PgridBase = loadMW - pvMW;

PimportBase = max(PgridBase,0);

baseCost = ...
    sum(buyPrice .* PimportBase * dt);

% ------------------------------------------------
% Initial population
% ------------------------------------------------

population = ...
    -PchMax + ...
    rand(popSize,N)*(PchMax+PdisMax);

bestHist = zeros(maxGen,1);

% ------------------------------------------------
% Evolutionary optimization
% ------------------------------------------------

for gen = 1:maxGen

    Jvec = zeros(popSize,1);

    for i = 1:popSize
```

```
        schedule = population(i,:);

        E = SOC0*Emax;

        energyCost = 0;
        penaltySOC = 0;

        for k = 1:N

            PB = schedule(k);

            % -----------------------------------
            % Update battery energy
            % -----------------------------------

            if PB < 0

                E = E + ...
                    etaCh*(-PB)*dt;

            else

                E = E - ...
                    PB*dt/etaDis;

            end

            SOC = E/Emax;

            % -----------------------------------
            % SOC constraint penalty
            % -----------------------------------

            if SOC < SOCmin

                penaltySOC = penaltySOC + ...
                    lambdaSOC*(SOCmin-SOC)^2;

            elseif SOC > SOCmax
```

```
        penaltySOC = penaltySOC + ...
            lambdaSOC*(SOC-SOCmax)^2;

    end

    % -----------------------------------
    % Grid power balance
    % -----------------------------------

    Pgrid = ...
        loadMW(k) - ...
        pvMW(k) - ...
        PB;

    Pimport = max(Pgrid,0);

    % -----------------------------------
    % Energy purchase cost
    % -----------------------------------

    energyCost = ...
        energyCost + ...
        buyPrice(k)*Pimport*dt;

end

% ----------------------------------------
% Final SOC penalty
% ----------------------------------------

SOCfinal = E/Emax;

penaltyFinal = ...
    lambdaFinal*(SOCfinal-SOC0)^2;

% ----------------------------------------
% Total objective
% ----------------------------------------
```

```
        Jvec(i) = ...
            energyCost + ...
            penaltySOC + ...
            penaltyFinal;

    end

    % ------------------------------------------------
    % Elitism
    % ------------------------------------------------

    [bestJ,bestIdx] = min(Jvec);

    elite = population(bestIdx,:);

    bestHist(gen) = bestJ;

    % ------------------------------------------------
    % New population
    % ------------------------------------------------

    newPopulation = zeros(size(population));

    newPopulation(1,:) = elite;

    row = 2;

    while row <= popSize

        % ----------------------------------------
        % Tournament selection: Parent 1
        % ----------------------------------------

        a = randi(popSize);
        b = randi(popSize);

        if Jvec(a) < Jvec(b)
            parent1 = population(a,:);
        else
            parent1 = population(b,:);
```

```
end

% ----------------------------------------
% Tournament selection: Parent 2
% ----------------------------------------

a = randi(popSize);
b = randi(popSize);

if Jvec(a) < Jvec(b)
    parent2 = population(a,:);
else
    parent2 = population(b,:);
end

% ----------------------------------------
% Crossover
% ----------------------------------------

if rand < Pc

    alpha = rand;

    child1 = ...
        alpha*parent1 + ...
        (1-alpha)*parent2;

    child2 = ...
        (1-alpha)*parent1 + ...
        alpha*parent2;

else

    child1 = parent1;
    child2 = parent2;

end

% ----------------------------------------
% Mutation
```

```
% ----------------------------------------

for k = 1:N

    if rand < Pm

        child1(k) = ...
            child1(k) + ...
            sigmaMut*randn;

    end

    if rand < Pm

        child2(k) = ...
            child2(k) + ...
            sigmaMut*randn;

    end

end

% ----------------------------------------
% Battery power limits
% ----------------------------------------

child1 = ...
    min(max(child1,-PchMax),PdisMax);

child2 = ...
    min(max(child2,-PchMax),PdisMax);

% ----------------------------------------
% Store children
% ----------------------------------------

newPopulation(row,:) = child1;

if row+1 <= popSize
    newPopulation(row+1,:) = child2;
```

```
        end

        row = row + 2;

    end

    population = newPopulation;

end

% ------------------------------------------------
% Final population evaluation
% ------------------------------------------------

Jvec = zeros(popSize,1);

for i = 1:popSize

    schedule = population(i,:);

    E = SOC0*Emax;

    energyCost = 0;
    penaltySOC = 0;

    for k = 1:N

        PB = schedule(k);

        if PB < 0

            E = E + ...
                etaCh*(-PB)*dt;

        else

            E = E - ...
                PB*dt/etaDis;

        end
```

```
        SOC = E/Emax;

        if SOC < SOCmin

            penaltySOC = penaltySOC + ...
                lambdaSOC*(SOCmin-SOC)^2;

        elseif SOC > SOCmax

            penaltySOC = penaltySOC + ...
                lambdaSOC*(SOC-SOCmax)^2;

        end

        Pgrid = ...
            loadMW(k) - ...
            pvMW(k) - ...
            PB;

        Pimport = max(Pgrid,0);

        energyCost = ...
            energyCost + ...
            buyPrice(k)*Pimport*dt;

    end

    SOCfinal = E/Emax;

    penaltyFinal = ...
        lambdaFinal*(SOCfinal-SOC0)^2;

    Jvec(i) = ...
        energyCost + ...
        penaltySOC + ...
        penaltyFinal;

end
```

```
[bestJ,bestIdx] = min(Jvec);

bestSchedule = population(bestIdx,:);

% ------------------------------------------------
% Simulate optimal schedule
% ------------------------------------------------

SOCbest = zeros(1,N+1);

SOCbest(1) = SOC0;

PgridBest = zeros(1,N);
PimportBest = zeros(1,N);

E = SOC0*Emax;

optimizedCost = 0;

for k = 1:N

    PB = bestSchedule(k);

    if PB < 0

        E = E + ...
            etaCh*(-PB)*dt;

    else

        E = E - ...
            PB*dt/etaDis;

    end

    SOCbest(k+1) = E/Emax;

    PgridBest(k) = ...
        loadMW(k) - ...
        pvMW(k) - ...
```

```
        PB;

    PimportBest(k) = ...
        max(PgridBest(k),0);

    optimizedCost = ...
        optimizedCost + ...
        buyPrice(k)*PimportBest(k)*dt;

end

saving = baseCost - optimizedCost;

% ------------------------------------------------
% PV self-consumption indicators
% ------------------------------------------------

gridImportEnergyBase = ...
    sum(PimportBase)*dt;

gridImportEnergyOpt = ...
    sum(PimportBest)*dt;

importReduction = ...
    gridImportEnergyBase - ...
    gridImportEnergyOpt;

% ------------------------------------------------
% Display results
% ------------------------------------------------

fprintf('Integrated PV-Battery Energy Management\n\n')

fprintf('Base cost without battery = %.2f EUR\n', ...
        baseCost)

fprintf('Optimized cost = %.2f EUR\n', ...
        optimizedCost)

fprintf('Cost saving = %.2f EUR\n\n', ...
```

```
        saving)

fprintf('Grid import without battery = %.3f MWh\n', ...
        gridImportEnergyBase)

fprintf('Grid import with battery = %.3f MWh\n', ...
        gridImportEnergyOpt)

fprintf('Grid import reduction = %.3f MWh\n\n', ...
        importReduction)

fprintf('Optimal Battery Schedule [MW]:\n')
disp(bestSchedule)

fprintf('SOC Trajectory:\n')
disp(SOCbest)

% ------------------------------------------------
% Convergence
% ------------------------------------------------

figure

plot(1:maxGen,bestHist,'LineWidth',1.6)

grid on

xlabel('Generation')
ylabel('Best Objective')

title('PV-Battery Energy Management Convergence')

% ------------------------------------------------
% Load and PV
% ------------------------------------------------

figure

plot(1:N,loadMW,'o-','LineWidth',1.6)
hold on
```

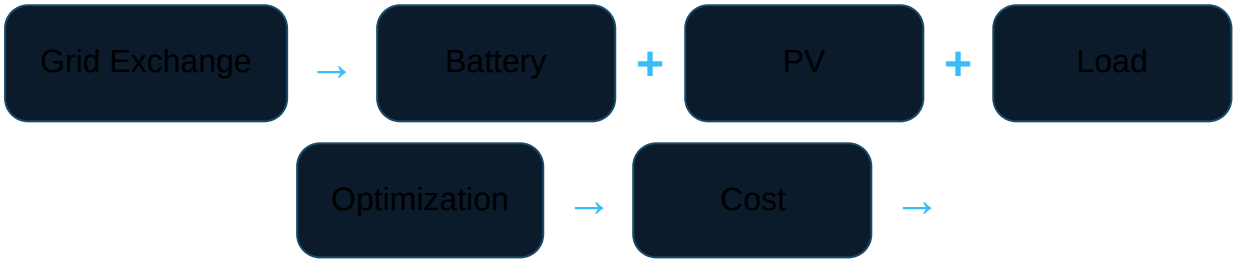
```
plot(1:N,pvMW,'s-','LineWidth',1.6)

grid on

xlabel('Hour')
ylabel('Power [MW]')

legend('Load','PV Generation', ...
       'Location','best')

title('Load and PV Profiles')

% ------------------------------------------------
% Battery schedule
% ------------------------------------------------

figure

bar(1:N,bestSchedule)

grid on

xlabel('Hour')
ylabel('Battery Power [MW]')

title('Optimal Battery Schedule')

% ------------------------------------------------
% SOC trajectory
% ------------------------------------------------

figure

plot(0:N,SOCbest,'o-','LineWidth',1.6)

hold on

yline(SOCmin,'--')
yline(SOCmax,'--')
```

```
grid on

xlabel('Hour')
ylabel('State of Charge')

title('Battery SOC')

% ---------------------------------------------------
% Grid exchange
% ---------------------------------------------------

figure

plot(1:N,PgridBase,'o-','LineWidth',1.6)
hold on

plot(1:N,PgridBest,'s-','LineWidth',1.6)

yline(0,'--')

grid on

xlabel('Hour')
ylabel('Grid Power [MW]')

legend('Without Battery','With Battery', ...
       'Location','best')

title('Grid Exchange Comparison')

% ---------------------------------------------------
% Electricity price
% ---------------------------------------------------

figure

stairs(1:N,buyPrice,'LineWidth',1.6)

grid on
```

```
xlabel('Hour')
ylabel('Price [EUR/MWh]')

title('Electricity Price')
```

17. البرنامج الكامل في MATLAB

يجمع البرنامج الحمل والطاقة الشمسية والبطارية والشبكة والسعر ضمن مسألة تحسين واحدة.



18. Interpreting Midday Operation

During high PV production:

$$PPV \approx Pload$$

or even:

$$PPV > Pload$$

charging the battery may increase local use of solar energy.

18. تفسير التشغيل في ساعات الظهر

عندما يكون توليد PV مرتفعًا، يمكن استخدام البطارية لتخزين الطاقة الشمسية بدل الاعتماد على الشبكة لاحقًا.

19. Interpreting Evening Operation

During high-price evening hours:

$$Price \uparrow$$

the stored solar energy can be discharged:

$$PB > 0$$

reducing expensive grid purchases.

19. تفسير التشغيل في ساعات المساء

عند ارتفاع السعر وانخفاض إنتاج PV، تستطيع البطارية استخدام الطاقة المخزنة لتقليل شراء الطاقة المرتفعة السعر.

20. PV Self-Consumption

PV self-consumption means using locally generated PV energy locally rather than exporting or curtailing it.

Battery storage can shift solar energy from the time of generation to the time of demand.

20. الاستهلاك الذاتي للطاقة الشمسية

تعني زيادة الاستهلاك الذاتي استخدام نسبة أكبر من الطاقة الشمسية محليًا.

تسمح البطارية بنقل الطاقة زمنيًا من ساعات الإنتاج الشمسي إلى ساعات الحاجة.

21. Experiment 1 — Remove PV

```
pvMW = zeros(size(pvMW));
```

Compare the battery schedule and energy cost.

21. التجربة 1 — إزالة PV

اجعل قدرة PV مساوية للصفر، ثم قارن طريقة تشغيل البطارية بالمسألة الأصلية.

22. Experiment 2 — Increase PV Size

```
pvMW = 1.5*pvMW;
```

Observe grid exchange and battery charging.

22. التجربة 2 — زيادة حجم نظام PV

زد قدرة النظام الشمسي ولاحظ تأثير ذلك على شحن البطارية والطاقة المستوردة من الشبكة.

23. Experiment 3 — Remove Battery

Set:

```
PchMax = 0;
PdisMax = 0;
```

This reproduces the PV-only reference case.

23. التجربة 3 — إزالة البطارية

ثبت قدرة الشحن والتفريغ عند الصفر للحصول على نظام PV دون تخزين.

24. Experiment 4 — Increase Battery Capacity

```
Emax = 3;
Emax = 5;
Emax = 10;
```

A larger battery may increase flexibility, but benefits depend on PV surplus, load profile, and electricity prices.

24. التجربة 4 — تغيير سعة البطارية

قارن الوفر الاقتصادي وتقليل الاستيراد عند استخدام سعات بطارية مختلفة.

25. Add Export Price

Define:

```
sellPrice = [20 20 20 20 20 20];
```

Then use:

$$Cost = BuyPrice \times Import - SellPrice \times Export$$

The optimal battery strategy can change significantly when exported PV energy has economic value.

25. إضافة سعر بيع للطاقة

عند وجود سعر لبيع الكهرباء الفائضة، تصبح الخوارزمية أمام قرار جديد: هل تخزن الطاقة أم تبيعها مباشرة؟

26. Add Battery Degradation

A more realistic objective can include:

$$Cdeg = cdeg \sum |PB(k)| \Delta t$$

Then:

$$J = GridCost + BatteryDegradationCost$$

26. إضافة تكلفة تدهور البطارية

يمكن جعل نموذج إدارة الطاقة أكثر واقعية من خلال احتساب تكلفة مرتبطة باستخدام البطارية.

27. Add PV Curtailment

When PV production is larger than local demand and storage capability, some PV power may need to be curtailed.

$$0 \leq Pcurt(k) \leq PPV(k)$$

The power balance becomes:

$$Pgrid = Pload - (PPV - Pcurt) - PB$$

27. إضافة خفض إنتاج PV

إذا كان إنتاج PV أكبر من قدرة الحمل والبطارية على الاستيعاب، قد نحتاج إلى خفض جزء من التوليد.

28. Multi-Objective Smart Energy Management

An advanced objective can combine:

$$J = w_1 Cost + w_2 GridImport + w_3 PVCurtailment + w_4 BatteryDe$$

The weights express the engineering priorities of the energy-management system.

28. إدارة طاقة متعددة الأهداف

$$J = w_1 \text{التكلفة} + w_2 \text{الاستيراد} + w_3 \text{خفض} PV + w_4 \text{تدهور البا}$$

تمثل الأوزان أولويات المصمم أو المشغل.

29. From Optimization to AI Control

The optimizer can generate many optimal operating examples:

$$\text{System State} \longrightarrow \text{Optimal Battery Action}$$

These examples can then train a neural network:

$$[\text{Load,PV,Price,SOC}] \longrightarrow \text{ANN} \longrightarrow \text{PB*}$$

This creates a bridge between offline optimization and fast AI-based real-time control.

29. من التحسين إلى التحكم بالذكاء الاصطناعي

يمكن استخدام الحلول المثلى الناتجة عن خوارزمية التحسين لتدريب شبكة عصبية تتعلم القرار الأمثل تقريبًا.

$$[\text{Load,PV,Price,SOC}] \longrightarrow \text{ANN} \longrightarrow \text{PB*}$$

وهذا يسمح لاحقًا بالحصول على قرار سريع جدًا في الزمن الحقيقي.

30. Lab Tasks

1. Enter load, PV, and electricity-price profiles.
2. Calculate the baseline grid exchange without battery.

3. Calculate baseline electricity cost.
4. Implement the battery SOC equations.
5. Implement the grid-power equation.
6. Add SOC and final-SOC penalties.
7. Run the optimizer.
8. Plot load and PV generation.
9. Plot the optimal battery schedule.
10. Plot SOC.
11. Compare grid exchange before and after optimization.
12. Calculate economic savings.
13. Calculate grid-import reduction.
14. Increase PV size and compare results.
15. Increase battery capacity and compare results.
16. Explain how this optimization can provide training data for an ANN.

30. مهام المختبر

1. أدخل منحنيات الحمل وPV والسعر.
2. احسب تبادل الشبكة دون بطارية.
3. احسب التكلفة المرجعية.
4. نفذ معادلات SOC.
5. نفذ معادلة قدرة الشبكة.
6. أضف عقوبات SOC وSOC النهائي.
7. شغّل المحسن.
8. ارسم الحمل وتوليد PV.
9. ارسم جدول تشغيل البطارية الأمثل.
10. ارسم SOC.
11. قارن تبادل الشبكة قبل التحسين وبعده.
12. احسب الوفر الاقتصادي.
13. احسب تقليل الاستيراد من الشبكة.
14. زد حجم PV وقارن النتائج.
15. زد سعة البطارية وقارن النتائج.

16. اشرح كيف يمكن استخدام هذه الحلول لتدريب ANN.

📝 Review Questions

1. Write the power-balance equation for the PV-battery-grid system.
2. What does positive battery power mean?
3. What does negative grid power mean?
4. Why can a battery increase PV self-consumption?
5. Why is the final-SOC condition important?
6. How does electricity price influence battery scheduling?
7. What is the effect of increasing PV capacity?
8. What is the effect of increasing battery capacity?
9. Why should battery degradation be included in realistic optimization?
10. How can optimal schedules be used to train an AI controller?

📝 أسئلة المراجعة

1. اكتب معادلة توازن القدرة لنظام PV والبطارية والشبكة.
2. ماذا تعني قدرة بطارية موجبة؟
3. ماذا تعني قدرة شبكة سالبة؟
4. كيف تزيد البطارية من الاستهلاك الذاتي للطاقة الشمسية؟
5. لماذا يعد شرط SOC النهائي مهمًا؟
6. كيف يؤثر سعر الكهرباء على جدولة البطارية؟
7. ما تأثير زيادة قدرة PV؟
8. ما تأثير زيادة سعة البطارية؟
9. لماذا يجب احتساب تدهور البطارية في النماذج الواقعية؟
10. كيف يمكن استخدام الجداول المثلى لتدريب متحكم ذكي؟