Recap → Electric system

Generator control → { primary → proportional: to quenche sudden $\delta\omega$ variation
secondary → to bring $\delta\omega = 0$, integral action
to steer back $\omega \to \omega_n$ }

(primary action): acts faster 10 ÷ 30s before pick
(secondary): some minutes to steer ω

synthesis method ⇝ 1, 2 gen case control synthesis

Electric systems – generation optimisation

- Generator cost modelling
- An introductory problem
- Optimisation background (constrain optimization)
- Global cost minimisation → simplest object
- Selfish utility maximisation → complementary problem, when bunch of Gen. maximize own quantity, not global
  in conflict with global optimiz
- Wrap-up

A system-level model

with to reference the thermo case (primary energy)

most convenient thermocase easy to threat

only automation costs...

- For simplicity we identify here cost and fuel consumption (i.e., we do not include plant maintenance, personnel and so on). (other cost NOT of our interest)
- (energy intake) Combustion is not equally efficient at all *plant loads*, i.e. – looking at the *thermal* load – for all values of p_c. you can define a curve

  we can define a curve representing specific consump.
- A *specific consumption* c_s ([kg of fuel per J], i.e., [kg/s of fuel per W]) is thus defined, which is typically a decreasing function of P_c in the admissible operation range, which normalised as done for p_c, in turn corresponds to an interval $(p_{c,min}, p_{c,max})$:

  (NOT sustained by gen.)
  - $p_{c,min}$ is the minimum "technical" load below which the generator cannot be operated, and may be something like 0.2–0.25, (even rarely 50%)
  - while $p_{c,max}$ is the maximum "guaranteed" power for the generator to work safely, and can be slightly greater than the unity, say 1.05–1.10, to accommodate for transient "exceptional" power releases to the network.

A system-level model

with reference to the thermo case

specific consumption (fuel to power ratio)

normalized combustion power

- Given the above, the fuel mass flowrate w_f and p_c are related by

P_c: power produced by combustion

$$w_f = c_s(p_c P_n)\, p_c P_n,$$

  which in our model we use to compute w_f while for this treatise considering p_c as the control input.
- In fact for control we consider the input to be θ_f and we assume this to be linearly related to p_c: as far as an estimate of $c_s(P)$ is available, however, the matter is no conceptual issue as the said estimated curve is invariantly smooth enough to be compensated for.

Generator cost models

- Most frequently, cost models are polynomial in the generated power and up to cubic. A typical form is

total cost rate as funct of gen. power P_{gi}

$$c_i(P_{gi}) = \underbrace{(k_{g1}P_{gi} + k_{g2}P_{gi}^2 + k_{g3}P_{gi}^3)}_{\text{fuel consumption}} k_F + k_{om0} + k_{om1}P_{gi}$$

cost of fuel per unit of energy

cost for operation + maintenance

other cost of consumables

where

c_i	$[€/h]$	is the cost rate,
P_{gi}	$[W]$	is the generated power,
k_{gj}, $j = 1\ldots3$	$[J/(hW^j)]$	are cost coefficients for pure generation,
k_F	$[€/J]$	is the fuel cost per unit of *energy*,
$k_{om\ell}$, $\ell = 0, 1$	$[€/(hW^\ell)]$	are cost coefficients for Operation & Maintenance.

- Note that, quite logically, the pure generation cost vanishes for $P_{gi} = 0$, while the O&M cost does not.

Generator cost models

- Why cubic?
- Because the fuel to generated power ratio $r_{fP}(P_g) = Q_f(P_g)/P_g$, where Q_f is the power yielded by fuel $[W]$ and P_g the generated power, thus making r_{fP} adimensional, typically has a minimum at the *optimal operating point* (by construction generally close to the maximum or *rated* power $P_{g,max}$). → [well approx by a parabolic behaviour]
- A good way to synthetically model this is to describe function $r_{fP}(P_g)$ as a *parabola*, specifying

you specify
  - the optimal (minimum) fuel to generated power ratio r^o_{fP},
  - the fraction p^o_g of $P_{g,max}$ corresponding to that optimal ratio, where p_g is defined as $P_g/P_{g,max}$,
  - and the fuel to generated power ratio r^{ml}_{fP} $(> r^o_{fP})$ at the minimum sustainable load, i.e., at $p^{ml}_g = P_{g,min}/P_{g,max}$,

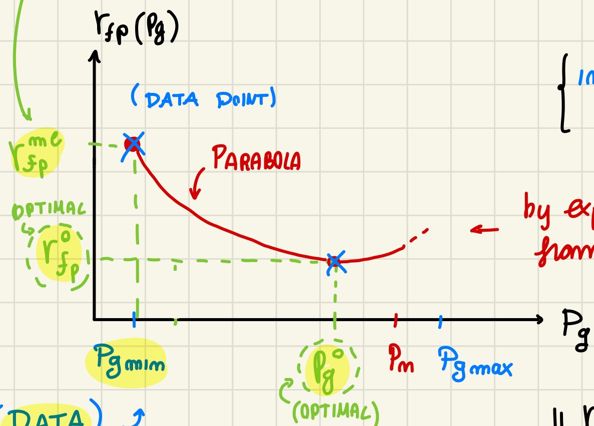
Why cubic cost models ?

defined as: (inverse of efficiency)

fuel to power ratio . $r_{fp} = \dfrac{\text{Kg of fuel}}{\text{Joules produced}} > 1$

by definition something is lost, you need to burn more than how much you produce!

[r_{fp} at minimum load "me"]



$\left\{ \begin{array}{l} \text{interval } (P_{g\,min}, P_{g\,max}) \\ P_{g\,max} \simeq 1.1\, P_m \text{ typically} \end{array} \right\}$

by experience we know from various data that the points of DATA are well interpolated by a PARABOLA

(DATA) (values highlighted)

you would like to stay here where you have the $\min\{r_{fp}\}$!

$\Downarrow$ $r_{fp}(P_g)$

typically in the form

$$r_{fp} = r_{fp}^{o} + K(P_g - P_g^{o})^2$$

{optimal value of r_{fp}}

(this "K" you compute from data) coefficient

You fit the data with offset r_{fp}^{o} and parameter P_g^{o}

$$r_{fp} = r_{fp}^{o} + K(P_g - P_g^{o})^2$$

Generator cost models

↳ to compulte the coeffcents in the plynomial $C_i(P_{gi})$ ↑ $K_{g1}, K_{g2}, K_{g3} \Rightarrow$ given by formulas

- The above computations give

$$r_{fP}(p_g) = r^o_{fP} + \frac{r^{ml}_{fP} - r^o_{fP}}{\left(p^o_g - p^{ml}_g\right)^2}\left(p_g - p^o_g\right)^2$$

thus

$$r_{fP}(P_g) = r^o_{fP} + \frac{r^{ml}_{fP} - r^o_{fP}}{\left(P^o_g - P_{g,min}\right)^2}\left(P_g - P^o_g\right)^2$$

Generator cost models

- Alternatively, $r_{fP}(P_g)$ can be obtained by interpolating experimental points (and again, a *parabola* normally suffices).
- Therefore, no matter how $r_{fP}(P_g)$ is obtained, $Q_f(P_g)$ can be expressed as

FUEL: fuel to Power

$$\text{cubic polynomial} \leftrightarrow \boxed{Q_f(P_g)} = \underbrace{r_{fP}(P_g)}_{\text{power ratio}} P_g = \left(\underbrace{r^o_{fP} + \frac{r^{ml}_{fP} - r^o_{fP}}{\left(P^o_g - P_{g,min}\right)^2} \left(P_g - P^o_g\right)^2}_{r_{fp}(P_g) \to \text{PARABULA}} \right) P_g$$

multiplyed by P_g become CUBIC POLINOMIAL!

that apparently contains powers of P_g from one to three, like we just wrote for the term $k_{g1}P_{gi} + k_{g2}P_{gi}^2 + k_{g3}P_{gi}^3$.

- In detail,

↳ $\left\{ \text{to compute the polynomial expression} \right\}$

$$k_{g1} = \frac{r^{ml}_{fP} P^{o\,2}_g - r^o_{fP} P_{g,min}\left(2P^o_g - P_{g,min}\right)}{\left(P^o_g - P_{g,min}\right)^2}, \quad k_{g2} = \frac{2P^o_g\left(r^o_{fP} - r^{ml}_{fP}\right)}{\left(P^o_g - P_{g,min}\right)^2},$$

$$k_{g3} = \frac{r^{ml}_{fP} - r^o_{fP}}{\left(P^o_g - P_{g,min}\right)^2}.$$

since

can be viewed as:

$$r_{fp} = \frac{\text{Kg/s of fuel}}{\text{Watts produced}} = \frac{W_F}{P_g} \implies \boxed{W_F = \underbrace{r_{fp}(P_g)}_{\text{parabula}} \cdot P_g}$$

(flow rate of fuel)

(generated power)

parabula · P_g

$\rightarrow$ so $W_F(P_g)$ is CUBIC !

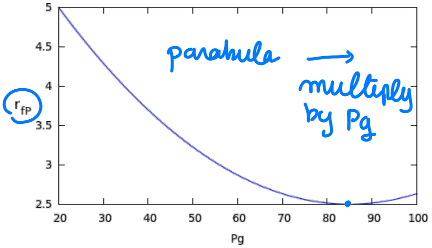
Generator cost models

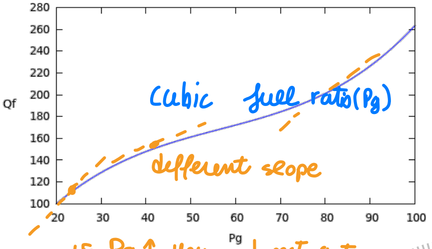
An example

IF you have lots of DATA you can use it to obtain more complex accurate models

machine learning ecc..

- The above cost model with $r^o_{fP} = 2.5$, $r^{ml}_{fP} = 5$, $P^o_g = 85$, $P_{g,min} = 20$ (supposing $P_{g,max} = 100$ for the plots) produces ⇓





you need a way to find opt. point!

IF Pg ↑ you don't get same fuel increment to provide

and, for completeness,

$$k_{g1} = 6.775, \quad k_{g2} = -0.101, \quad k_{g3} = 5.917 \cdot 10^{-4}.$$

- Note: r_{fP} can be interpreted as the *inverse* of the fuel-to-power *efficiency* $\eta_{fP}(P_g) = P_g/Q_f(P_g)$, ranging in this case from 0.4 (optimal point) to 0.2 (minimum sustainable load).

Problem statement

remember we can have gen. of different sizes

Preliminaries

better NOT to reason with norm quant

- To understand, better *not* to reason with normalised quantities.
- Consider a network with N generators: at any given moment, the total generated power must equal the electric power demand, i.e.,

{ to provide this... }

total generated

$$\sum_{i=1}^{N} P_{gi} = P_e$$

network demand

- Primary and secondary control can ensure this, and also keep frequency to the set point (if the power request is feasible, of course). → keeping $\omega = \omega_n$
- But as said, what about cost?
- Knowing the efficiency curve of each generator, one can write N functions to relate each P_{gi} $[W]$ to a "cost rate" c_i $[€/s]$ or $[€/h]$.
- Of course each $c_i(P_{gi})$ is a monotonically increasing function and each generator has limits, i.e., $P_{gi,min} \leq P_{gi} \leq P_{gi,max}$.

from Q_f : [kg/s]/[J] fuel consumption, multiplied by cost of fuel → [€/s] COST RATE fuel flow rate

from $W_f = \left[\frac{kg}{s}\right]$ (multiply by fuel cost $[€/kg]$)

$$\frac{kg}{s} \cdot \underbrace{\frac{€}{kg}}_{K_f} \Rightarrow \frac{€}{s}$$ cost rate

$\{$ still cubic $\}$

how much spent per second to produce!

$$\boxed{C = W_F \cdot K_F}$$ = cost rate

:= cost of one unit of fuel

still cubic function!

minimizing the derivative ∀t ! lead to min of integral

= min fuel consumption = min total fuel flowrate

Problem statement

Cost function

- Supposing for this first case that the purpose is to minimise the overall cost, it can be stated as that of minimising the overall cost rate, hence as

↳ min overall flowrate!

constrained optimiz problem ← constraints

$$
\begin{array}{ll}
\min \sum_{i=1}^{N} c_i(P_{gi}) & \\
\text{s.t.} \quad \sum_{i=1}^{N} P_{gi} = P_e, & \\
\qquad P_{gi,min} \leq P_{gi} \leq P_{gi,max}, \; i = 1 \ldots N. &
\end{array}
$$

equality constrain (respect demand)

inequality constraints on gen power range

- Note: the problem can be much more complex as *sets* of generators may aim for example at minimising *their* cost while together generating a stipulated total power or a given share of P_e, or at maximising their revenue irrespectively of the rest, and so forth.
- We shall face some additional complexity later on, compatibly with the available space.
- For the moment, let us understand the principles.

An introductory case

principles

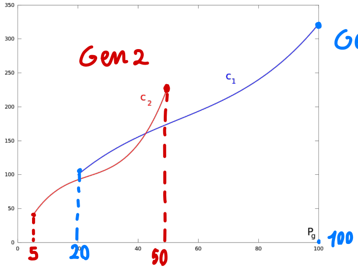
Two generators

Given 2 generators:

- Generator data:

$$\begin{cases} P_{g1,max} = 100 & P_{g1,min} = 20 & P^o_{g1} = 80 & r^o_{fP1} = 3 & r^{ml}_{fP1} = 5 \\ P_{g2,max} = 50 & P_{g2,min} = 5 & P^o_{g2} = 35 & r^o_{fP2} = 3.5 & r^{ml}_{fP2} = 8 \end{cases}$$

{cost curves} $c_i(P_g)$



NET DEMAND

- Network power demand: $P_{e,max} = 130$, $P_{e,min} = 10$. $P_e \in 10 \div 130$

- We suppose that generators can be activated/deactivated at any time (this impacts operation scheduling, not in our scope). (not always realistic)

sometimes NOT available gen...

An introductory case

Two generators

to determine the forecast $\widehat{P}_e$ is determined by trend on daily data ecc..

typically over next 30min

- Basic idea (which is general w.r.t. the example):
  1. take the forecast power demand $\widehat{P}_e$ for the next "period" (day, hour,...),
  2. determine the optimal generation distribution $\{P_{gi}^{opt}\}$ yielding $\widehat{P}_e$ at minimum cost, → solve the optimiz. problem
  3. send each of the so obtained generation requests P_{gi}^{opt} to the corresponding generator as a bias value, ↳ result is: P_{gi}: request of each gen → send this as BIAS for next period (tertiary control)
  4. and let primary and secondary control act as usual.
     ↳ to match the real request respect the forecast (error)

  result

- Let us now concentrate on the $\widehat{P}_e \mapsto \{P_{gi}^{opt}\}$ problem, other aspects (to the feasible extent) later on.

how to find the BIAS to use as input for the Gen.

how to send the BIAS to give to a gen. for next period $\widehat{P}_e$!

An introductory case

Example

Two generators

- Consider all the generator combinations, and determine the minimum and maximum power that can be generated by each of them:

POSSIBLE COMBINATIONS:

Both G1 & G2 simultaneously

$$G1 \to [20, 100], \quad G2 \to [5, 50], \quad G1 + G2 \to [25, 150].$$

only G1 — only G2 — Σmin, Σmax

- Consequently, divide the $\widehat{P}_e$ range in intervals I_i and determine the feasible combinations for each of the said intervals (easier to show than to explain):

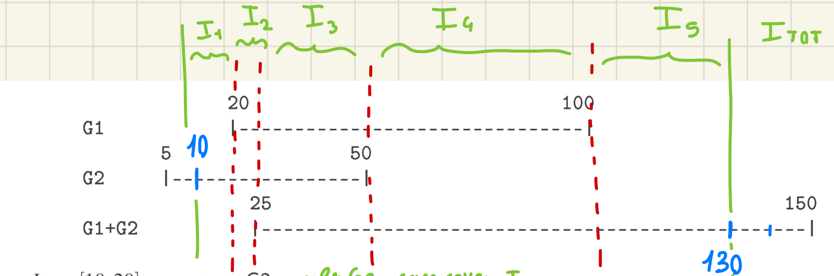
each possible combination

divide Power interval in chunks to check which genare feasible to use!

I_1	$[10, 20]$	$\to$	G2
I_2	$[20, 25]$	$\to$	G1,G2
I_3	$[25, 50]$	$\to$	G1,G2,G1+G2
I_4	$[50, 100]$	$\to$	G1,G1+G2
I_5	$[100, 130]$	$\to$	G1+G2

to check which usable on each range ⇝ by partitioning on gen, you we know that $\boxed{P_c \in 10 \div 130}$ from EXAMPLE DATA

you divide power interval in chunks and ∀ chunk identify which G_i combination is feasible!



I_1	[10, 20]	→	G2	only G_2 can cover I_1
I_2	[20, 25]	→	G1, G2	only singular G_1 or G_2, not G_1&G_2
I_3	[25, 50]	→	G1, G2, G1+G2	all cases
I_4	[50, 100]	→	G1, G1+G2	
I_5	[100, 130]	→	G1+G2	

↑ here there are possible combination where only one Gen. is enough → NOT usual in real cases! ⇒ in other case when more Gen acts toghether G_1&G_2

{ you have to decide how much of G_1 and of G_2 you use!

solve an optimiz problem!

An introductory case

Two generators

- For combinations with more than one generator, the optimal distribution (minimum total cost) has to be found.
- With only two generators the only case to consider is G1+G2, and we can proceed by substitution (we shall see something more general later on):

since $P_{g1}+P_{g2}=\widehat{P}_e$ must manage overall request

by definition

$$P_{g2}=\widehat{P}_e-P_{g1} \Rightarrow \boxed{c_{12}(P_{g1})=\underbrace{c_1(P_{g1})}_{\text{GEN1 cost}}+\underbrace{c_2(\widehat{P}_e-P_{g1})}_{\text{GEN2 cost}}}$$

only free var

cost of the combination G1&G2 depends only on P_{g1}

- Then we take the first and second derivative of $c_{12}(P_{g1})$ w.r.t. the only remaining independent variable P_{g1},
- find a possible minimum cost $c_{12}^{opt}(\widehat{P}_e)$ inside the power range of both generators (otherwise the minimum is at one of the two distribution extrema),
- determine the power distribution – i.e., $P_{g1}^{opt}(\widehat{P}_e)$ – corresponding to the minimum.

An introductory case

Two generators

done here for different values of power demand $\widehat{P}_e$

2 demands: $\begin{cases} \widehat{P}_e = 70 \\ \widehat{P}_e = 100 \end{cases}$

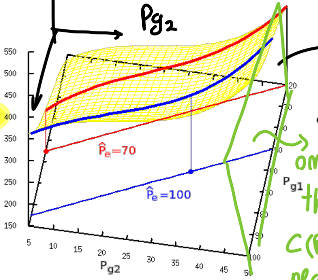
- Graphical interpretation for the G1+G2 combination:

C_{12}

P_{g2}

P_{g1}

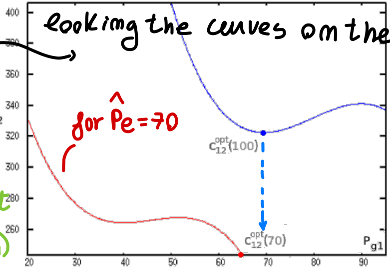
(plotted cost function (yellow one...)



on that $C(P_{g1})$ plane

looking the curves on the plane parallel to P_{g1} !

for $\widehat{P}_e = 70$



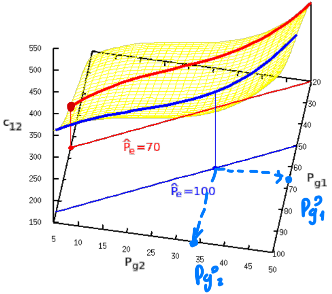
- As can be seen, for a given $\widehat{P}_e$, the optimal distribution can be inside the segment of the $P_{g1} + P_{g2} = \widehat{P}_e$ straight line, or at one of its extrema.

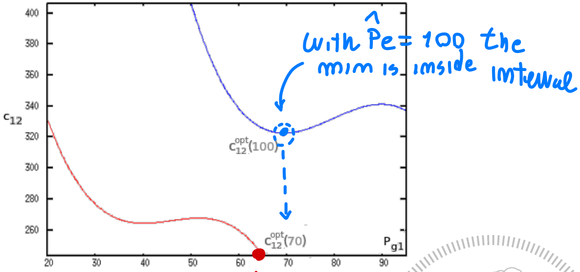
An introductory case

Two generators

(by limitation) ⇐ optimal power P_g inside the range or in an extreme of the curve

- Graphical interpretation for the G1+G2 combination:



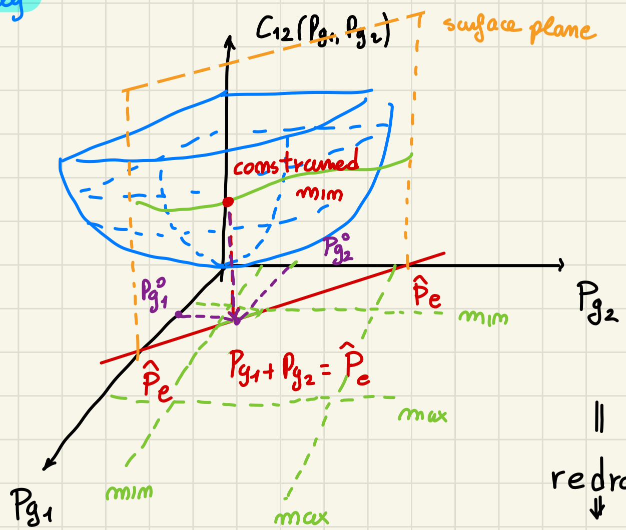


- As can be seen, for a given $\widehat{P}_e$, the optimal distribution can be inside the segment of the $P_{g1} + P_{g2} = \widehat{P}_e$ straight line, or at one of its extrema.

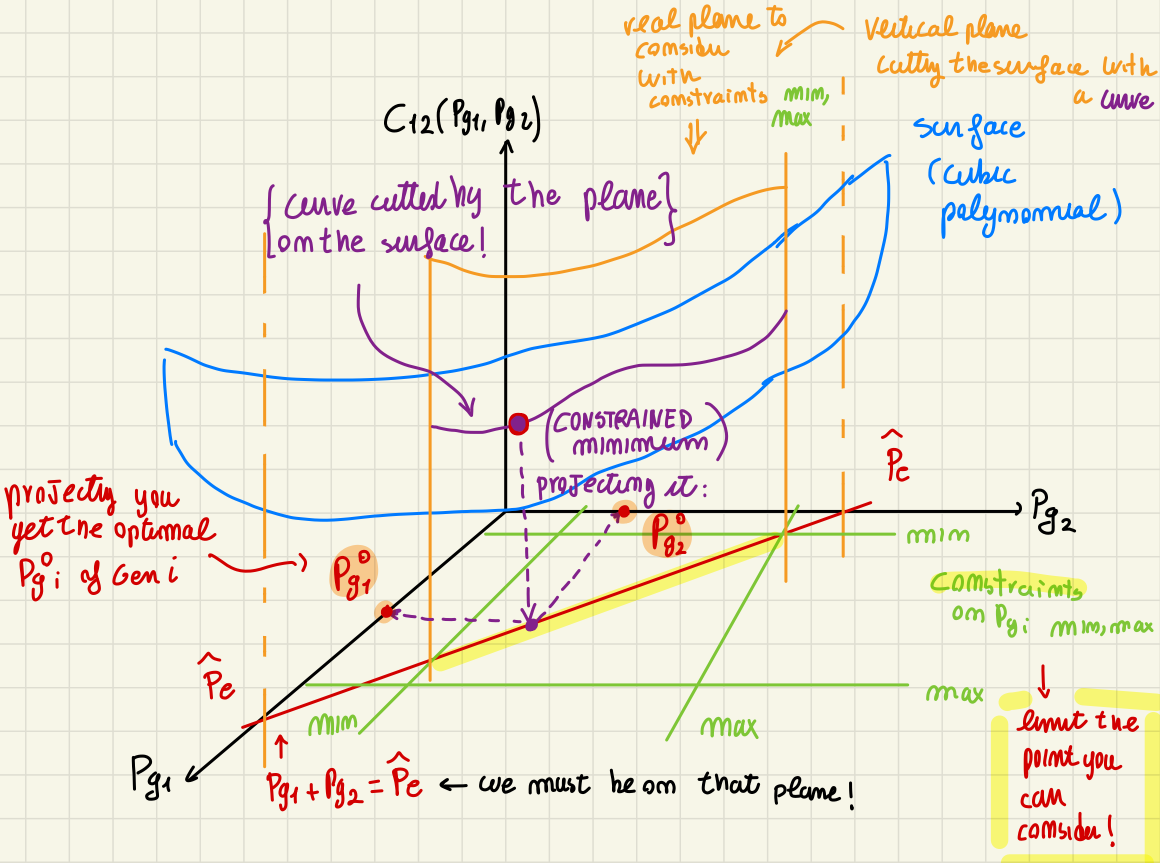
the min can be NOT a stationary point! (cutted by feasibility interval!)

GRAPHICAL interpretation

Graphically



||
redraw
⇓



An introductory case

Two generators

all toghether...

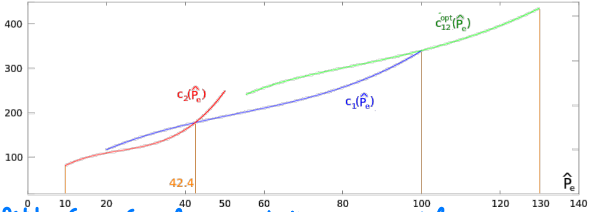
Im each case when you have more than one gen. you have a ranking...

- As illustrated by the previous graphical interpretation, for each value of $\widehat{P}_e$, the combination can now be chosen that provides the minimum cost; this is shown below:

G1
G2
G1+G2

feasible combination

you check.. what is min achievable by G1+G2? what Cost by G1 or G2 alone → take min possible



for each possible combination you deteme min cost → and the one with smallest min cost WIN

- Finally, based on the choice just made, the bias (or "tertiary control") values $P_{b1,2}$ for $P_{g1,2}$ are determined.
- Let us now quit this introductory example and naïve technique, and move toward establishing a methodology.

easy with 2 Gen, but with more possible combinations of Gi you need math metodology!

- Before, however, we need to review some mathematics.

you chose the combination with min feasible cost

↓

chose the one with min optimal cost → through metodology

1. intervals I_i of G_i combination feasible
2. $\forall$ intervals feasible combination (knowing forecast)
3. optimize each combin.

↓

4. choose the one with smallest optimum!

Optimisation background

Constrained optimisation – Lagrange multipliers – Karush-Kuhn-Tucker (KKT) equations

Problem statement

- We want to minimise a real function f of N_x real variables x_i, i.e.,

$$f(x_1, x_2, \dots x_{N_x}), \qquad f(\cdot,\cdot,\dots\cdot) \in \Re, \; x_i \in \Re, \; i = 1 \dots N_x,$$

subject to N_e equality constraints in the form $(\Sigma P_{gi} = P_e)$

$$g_i(x_1, x_2, \dots x_{N_x}) = 0, \qquad g_i(\cdot,\cdot,\dots\cdot) \in \Re, \; i = 1 \dots N_e,$$

and to N_i inequality constraints in the form $(P_{gi\,min} \leq P_{gi} \leq P_{gi\,max})$

$$h_i(x_1, x_2, \dots x_{N_x}) \geq 0, \qquad h_i(\cdot,\cdot,\dots\cdot) \in \Re, \; i = 1 \dots N_i.$$

- *Caveat:* this is *not* a math lecture. We shall assume that "everything is regular enough", and not even mention several hypotheses that would be necessary for a rigorous treatise. ~ derivable ecc.. integrals exists...
  necessary hyp! regularity

Optimisation background

Only equality constraints – Lagrange multipliers

$\boxed{g_i(\ldots)=0}$ constraint!

1 • Form the problem's *Lagrangian* as

(constrain function) "$g_i(\cdot)$"

$$L = f(x_1, x_2, \ldots x_{N_x}) + \sum_{i=1}^{N_e} \lambda_i g_i(x_1, x_2, \ldots x_{N_x})$$

L = function to minimize + Σ_i (all equality constrain func) $* \lambda_i$

introducing N_e additional real unknowns λ_i, named the *Lagrange multipliers*.

• Compute the gradients of L w.r.t. vectors $x = [x_1 \ldots x_{N_x}]' \in \Re^{N_x}$ and $\lambda = [\lambda_1 \ldots \lambda_{N_e}]' \in \Re^{N_e}$, i.e.,

respect decision var. | respect lagrange multiplyer

GRADIENT
∇:= row vector with scalar derivative

$$\nabla_x L(x,\lambda) = \left[\frac{\partial L}{\partial x_1} \frac{\partial L}{\partial x_2} \ldots \frac{\partial L}{\partial x_{N_x}}\right], \quad \nabla_\lambda L(x,\lambda) = \left[\frac{\partial L}{\partial \lambda_1} \frac{\partial L}{\partial \lambda_2} \ldots \frac{\partial L}{\partial \lambda_{N_e}}\right],$$

having respectively N_x and N_e (function) components.

respectively

$\nabla_x L \in (\mathbb{R}^{N_x})^T$ $\quad \nabla_\lambda L \in (\mathbb{R}^{N_e})^T$

Optimisation background

Only equality constraints – Lagrange multipliers

Remark!

where x_k appear of $f(\cdot)$ and also inside all $g_i(\cdot)$ constraint functions

- Observe that the k-th component of $\nabla_x L$ is $= \frac{\partial L}{\partial x_k}$

$$\frac{\partial L}{\partial x_k} = \frac{\partial f}{\partial x_k} + \sum_{i=1}^{N_e} \lambda_i \frac{\partial g_i}{\partial x_k} = \frac{\partial f}{\partial x_k} + \lambda' \cdot \begin{bmatrix} \frac{\partial g_1}{\partial x_k} \\ \vdots \\ \frac{\partial g_{N_e}}{\partial x_k} \end{bmatrix}$$

$\lambda' = (\lambda_1 \;\; \lambda_2 \dots \lambda_{N_e})$

x_k is on $f(\cdot)$ and all the $g_i(\cdot)$ constraint functions

where "$\cdot$" denotes the scalar product. Therefore

Overall..

$$\nabla_x L = \nabla_x f + \lambda' \underbrace{\overbrace{\begin{bmatrix} \frac{\partial g_1}{\partial x_1} & \cdots & \frac{\partial g_1}{\partial x_{N_x}} \\ \vdots & & \vdots \\ \frac{\partial g_{N_e}}{\partial x_1} & \cdots & \frac{\partial g_{N_e}}{\partial x_{N_x}} \end{bmatrix}}^{N_x \text{ columns}}}_{J_x g} = \nabla_x f + \lambda' \cdot \begin{bmatrix} \nabla_x g_1 \\ \vdots \\ \nabla_x g_{N_e} \end{bmatrix} = \nabla_x f + \lambda' J_x g$$

N_e rows

where $J_x g$ is the Jacobian of the constraints g w.r.t. x. (Jacobian matrix of g respect x)

- Also, observe that the k-th component of $\nabla_\lambda L$ is g_k.
  (by construction)

Kth component of $\nabla_\lambda L = g_K$ $\rightsquigarrow$ in fact

from $\nabla_\lambda L = \frac{\partial L}{\partial \lambda} = \left[\frac{\partial L}{\partial \lambda_1} \cdots \frac{\partial L}{\partial \lambda_{Ne}} \right]$

with λ_i only on

$$L = \cancel{f(x)}^{0} + \sum_i g_i(x)\lambda_i \rightsquigarrow \frac{\partial L}{\partial \lambda_K} = g_K(x)\ !$$

$\partial/\partial\lambda_i$ $\qquad$ $\partial/\partial\lambda_i$

Optimisation background

Only equality constraints – Lagrange multipliers

$\nabla_\lambda L = 0 \Leftrightarrow g_k = 0 \ \forall k$ so respected constraints!

- Now, suppose that (x^o, λ^o) is a solution for the system of $N_x + N_e$ equations

set to 0 all the components

that x^o, λ^o solve these eqs.

$$\begin{cases} \nabla_x L(x,\lambda) &= 0_{1\times N_x} \\ \nabla_\lambda L(x,\lambda) &= 0_{1\times N_e} \end{cases}$$

$\frac{\partial L}{\partial \lambda}$ are the equality constrain function → if = 0 OK satisfied

in the $N_x + N_e$ unknowns (x, λ), termed the Lagrangian Multipliers (LM) equations.

- (The second equation says that x^o fulfils $g(x)$) (For the first we have two cases)

2 cases for the 1st equation

- Case 1: $\nabla_x f$ in x^o is a zero vector. → equations become $\lambda^T J_{xg} = 0$

1

  - In this case x^o is a *stationary point for $f(x)$ independently of the constraints $g(x)$*.
  - In addition, given the expression of $\nabla_x L$, the equation reduces to $\lambda' J_x g = 0$; since (x^o, λ^o) fulfils it, either λ^o is a zero vector, or the gradients $\nabla_x g_i, i = 1 \ldots N_e$ evaluated in x^o are linearly dependent. → lin comb of gradient is 0 → so dependent! meaning that redundant constraints

meaning that there are redundant constraints

  - If the $\nabla_x g_i$ are linearly dependent in x^o we shall then say that x^o *may* not be a *regular point* for the constraints g; we do not further discuss this matter.

to salve $\nabla_x L(x,\lambda) = \underline{0}_{1\times N_x}$ we have 2 cases..

remembering →

$$\nabla_x L = \nabla_x f + \lambda' J_x g$$

f gradient + matrix vector product

↓

case (1):

IF $\underbrace{\nabla_x f = 0}$ ⇝ simply the equation becomes

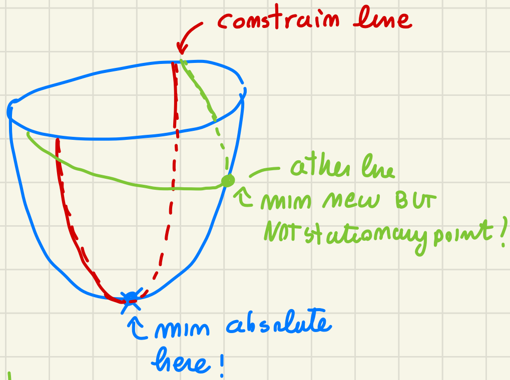
↓

that x is a stationary point independently of constraints

$$\lambda' J_x g = \underline{0}_{1\times N_x}$$

↳ for example: to minimize:

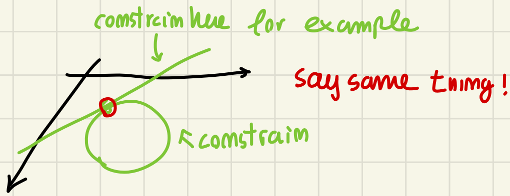
here that constraint line is useless! ↜ because that point is min independently from constraints...



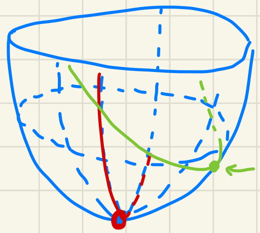
While of the line of constraim is another! ⤵

new min NOT of Global surface! BUT of constraint

If linear dependent $\nabla_x g_i$ ⇝ redundant constraints



for EXAMPE to minimze



min !

curve min but NOT of the overall system → STATIONARY point

Optimisation background

Only equality constraints – Lagrange multipliers

meaning that (to solve Lag. equat.)!

($\lambda' J_x g \neq 0$ also!)

more interest case...

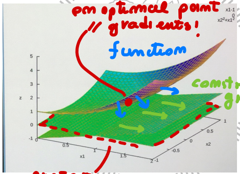
When $\nabla_x f \neq 0$ in x^o

When our x^o min is not a stationary point of the surface but only of curve

- Case 2: $\nabla_x f$ in x^o is not a zero vector.
  2
  - In this case λ^o cannot be a zero vector either, or the considered equation $\nabla_x f + \lambda' J_x g = 0$ cannot be satisfied (contrary to the hypothesis).
  - Also, rewritten as $\nabla_x f = -\lambda' J_x g$, the same equation says that the gradients of $f(x)$ and $g(x)$ w.r.t. x are parallel in x^o. (parallel gradients of f and g)!
  - Let now $z^o = f(x^o)$, and consider the hypercurve in $\Re^{N_x+1}$ obtained by intersecting the hypersurfaces $z = f(x)$ and $g(x) = 0$. Moving on that hypercurve away from (x^o, z^o), that apparently belongs to it, *locally* produces no variation of z. Therefore, x^o is a (local) stationary point for $f(x)$ constraine by $g(x) = 0$.
  - Since this may be hard to grasp, let us see an example with Maxima:

```
f : x1^2+x2^2;                                  funct to min
g : x1-1;                                       constrain      /* grad_x g = [1 0]         */
L : f+lam*g;                                    LAGRANGIAN
solve([diff(L,x1),diff(L,x2),
       diff(L,lam)],[x1,x2,lam]);               } solve the syst of eq → x1=1, x2=0, λ=-2
plot3d([f,g,0, [x1,0,2], [x2,-1,1]]);           → plotting in 3D (f,g,0 for x1,x2) ⇒
subst([x1=1,x2=0],jacobian([f],[x1,x2])); /* grad_x f || grad_x g in xo */
subst([x1=1,x2=1],jacobian([f],[x1,x2])); /* and not e.g. here          */
```

↳ simulating that in maxima ⇒



$\nabla_x f = -\lambda' J_x g \Rightarrow$ "the GRADIENT are PARALLEL"

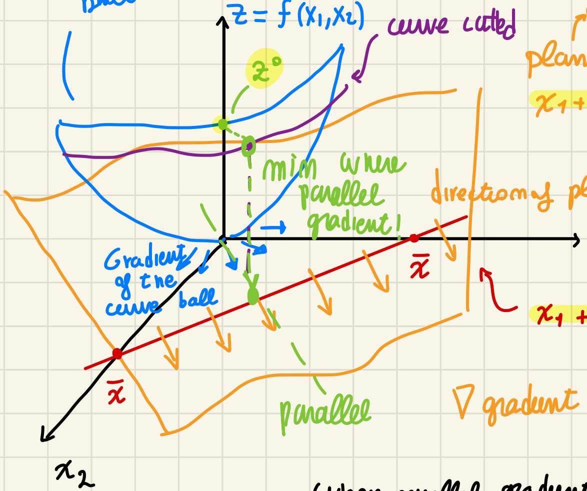
f to be minimized s.t $g(x_1,x_2)=0$ with $g = x_1 + x_2 - \bar{x}$

function to minimize! "ball" Gradient away from origin

↷ max increase direction on it is the mooving away from the origin

(plane cutting through)

plane: (inclined plane, not "vertical") $x_1+x_2-\bar{x}=0$ viewed in 3D



constraint $x_1 + x_2 - \bar{x} = 0$ viewed in 2D

∇ gradient directions of the plane (all parallels)

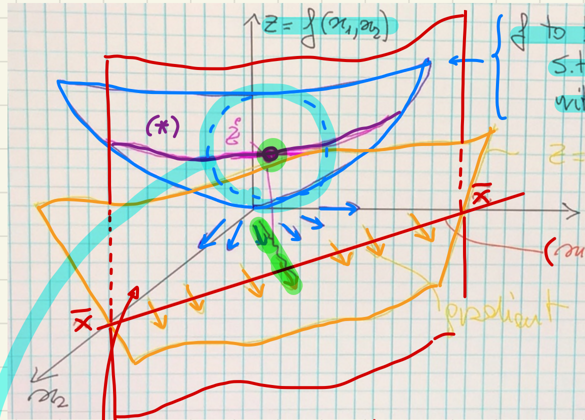
When parallel gradient you get the min!

When the gradient of the ball and of the plane are parallel ⇝ we get the min

↓

more since ortogonally to gradient ⇝ when no more increment possible... (no variation)

since the gradient of the constraint function and of the ball f to min are parallel → mooving ortogonally to the position where ∇ parallel I get no increment! (no variation)



$\left\{ \begin{array}{l} f \text{ to be minimised} \\ \text{s.t. } g(x_1, x_2) = 0 \\ \text{with } g = x_1 + x_2 - \bar{x} \end{array} \right\}$

(x_1, x_2, f) plane

$z = x_1 + x_2 - \bar{x}$ VIEWED IN 3D

↳ $z = g(x)$: inclined plane!

$(x_1 + x_2 - \bar{x} = 0)$ VIEWED IN 2D

↳ $g(x) = 0$ trace of constrain plane on X_1, X_2 PLANE

rising a vertical plane from $X_1 + X_2 - \bar{X} = 0$...

$g(x) = 0$ VIEWED IN 3D → become a plane! $(\forall z)$

↓ and this plane in 3D → $g(x)$ cuts an hypercurve with the surface! (*)

↳ (3D CURVE) (by plane and solid interseption)

ZOOM (*)(*)

plane gradient lays in 2D ORTOGONAL to plane $\partial/\partial x_1$, $\partial/\partial x_2$ always parallel → lays on X_1-X_2 plane ⇓ direction where the plane Grow!

↙ ↓ ↘ blue vectors are the max Growth of $f(x_1, x_2)$ radially directed!

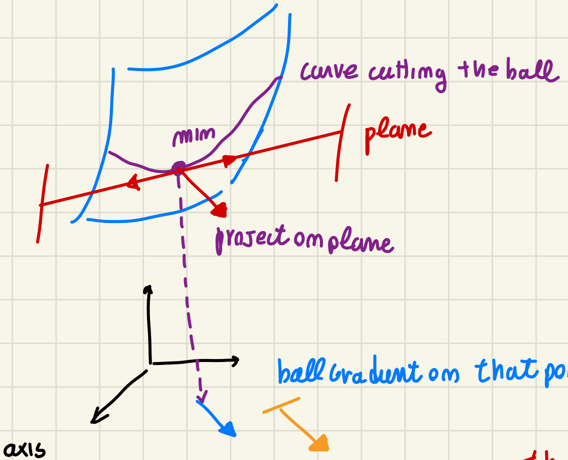
↳ so when reached the min point...

plane and ball has parallel gradient (as equations says)

↳ point where the two are parallel → min!

(*)(*) ZOOM on that area.. ⇒

ZOOM... here we notice part of f surface



considering the tangent due to constraint plane

ball gradient on that point // to constraint gradient

on that direction along constraint I move $\perp$ to gradient
↳ locally NO VARIATION!

here the gradient of the blue surface is $\perp$ to magenta line tangent on red plane... ← { because the product of my motion to gradient is 0 scalar (mooving along // line)

$\nabla_x f, \nabla_x g$ parallel in x_0

$z^\circ = f(x^\circ)$ hypercurve in $\mathbb{R}^{n+1}$

↑ interseption of $\begin{cases} z = f(x) & /// \\ g(x) = 0 & /// \ 3D \end{cases}$

mooving on that curve → locally NO VARIATION.. I moove $\perp$ to gradient
⇓
STATIONARY POINT!

Optimisation background

Only equality constraints – Lagrange multipliers

simple operationally:
1. form the Lagrangeant
2. compute ∂x, ∂λ
3. set all to 0
→ 4. solve problem
5. get candidates of optimal point constrained

v

- Conclusion: x^o fulfils the LM equations
  $\Rightarrow$ it is a *candidate* constrained optimal point.
- We have then found a set of *necessary*, *first-order* conditions for *local* constrained optimality.
- Further studies could e.g. study via Hessian analysis whether is x^o a minimum, a maximum, or neither. ↳ to checking min/max... studying hessian matrix (II derivative) → max/min ecc..
- For our purposes we can stop here, however we need to address *inequality* constraints.

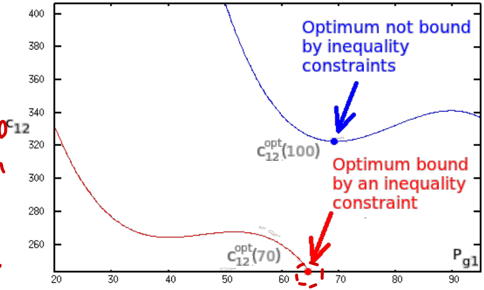
↳ how to deal with it! : to deal with inequality we moove from Lagrangean Multiplyer to KKT equations! ⟹

Optimisation background

Equality and inequality constraints – Karush-Kuhn-Tucker (KKT) equations

- Major difference w.r.t. the equality-only (LM) case: a solution may *not* be a stationary point, as some inequality constraints may bind it. (the inequality can bind stationary solution)
- Observe that we have already found such a case in the introductory example with two generators:

for $\hat{P}_e = 70$ the optimum is binded @ the extremum!



- Hence the LM equations cannot be used here (i.e., the Lagrange *rationale* still works, but we need to introduce some modifications).

Optimisation background

Equality and inequality constraints – Karush-Kuhn-Tucker (KKT) equations

- We form again the Lagrangian, this time however in the form

{inequality constraint function : $h_J(x)$}

inequality constraints defined by $h_J(x) \geq 0$

Extended Lagrangian

add the part related to inequality constr function by μ_J

$$
\begin{aligned}
L(x,\lambda,\mu) &= f(x) + \sum_{i=1}^{N_e} \lambda_i g_i(x) + \sum_{j=1}^{N_i} \mu_j h_j(x) \\
&= f(x) + \lambda \cdot g(x) + \mu \cdot h(x)
\end{aligned}
$$

where another multiplier vector $\mu = [\mu_1 \dots \mu_{N_i}]' \in \Re^{N_i}$ is introduced w.r.t. the case with equalities only, and $h \in \Re^{N_i}$ is the vector of functions h_j,

- and give the following simple definitions:
    - if at a certain point x^o a certain inequality constraint h_j is satisfied with equality – i.e., if $h_j(x^o) = 0$ – we shall say that the constraint h_j is *active* (or *binding*) in x^o; ← hit boundary (we hit the boundaries)
    - if the constraint is satisfied with the $>$ sign – i.e., if $h_j(x^o) > 0$ – we shall say that it is *inactive* (or *nonbinding*) in x^o; (not hit boundaries)
    - otherwise (obviously) the constraint is *violated* in x^o.

outside valid region

Optimisation background

Equality and inequality constraints – Karush-Kuhn-Tucker (KKT) equations

(nothing bind or violated)

means that x^o would be solution without inequality constr!

- Now (we again look for necessary conditions) suppose that x^o is a solution (i.e., an optimal point) with neither binding nor violated inequality constraint, i.e., that $h_j(x^o) > 0 \forall j$.

↑ SOLUTION for the problem! inside valid region NOT on boundary

verify NO violation of constraint

- In this case x^o is also a solution for the LM problem, as setting $\mu = 0$ makes the term $\mu \cdot h(x^o)$ contribute zero to L. $\Rightarrow$ BUT setting $\mu=0$, we catch also sol with violated constraint $\rightarrow h \gtrless 0$ need to check

- Note that also a solution for the LM problem *violating* some inequality constraint would fall in the same case, but it is not difficult to see if said constraints are violated or nonbinding. In the following we assume that such a *feasibility check* is always performed.

$\Rightarrow$

- The most interesting case is when at least one inequality constraint is binding. Let us expand a bit on this. !

trick...

Optimisation background

Equality and inequality constraints – Karush-Kuhn-Tucker (KKT) equations

trick: multiply each equation by μ,

- By adopting the same notation introduced in the LM problem, consider the system of $N_x+N_e+N_i$ equations → form a syst of eq...

$$\begin{cases} \nabla_x L(x,\lambda,\mu) & = & \nabla_x f(x)+\lambda' J_x g(x)+\mu' J_x h(x) & = & 0 & (N_x) \\ \nabla_\lambda L(x,\lambda,\mu) & = & g(x) & = & 0 & (N_e) \\ \mu' \circ \nabla_\mu L(x,\lambda,\mu) & = & \mu' \circ h(x) & = & 0 & (N_i) \end{cases}$$

↑ element by element multiplication

in the $N_x+N_e+N_i$ unknowns (x,λ,μ), where "$\circ$" denotes the Schur (element by element) product. -> with the trick of multiply each equation by μ! → threate ineq. constr as eq. constrain only when binding

- Roughly speaking, a solution bound by some inequality constraints will satisfy an LM problem where the said constraints are fictitiously treated as equality ones (whence the last term in the first equation) provided that only those (binding) constraints are actually accounted for, which is ensured by the third equation (of course s.t. the necessary feasibility checks).

↳ IF NOT BINDING, $h(x) \neq 0$ so fix $\mu' = 0$ OK
IF BINDING, $h(x) = 0$, μ' whatever value OK ✓

TRICK: treat inequality constraints as equality when binding

$$\mu' \circ h(x) = 0$$

this can be $\neq 0$ even when bind

$= 0$ when binding ✓

only binding constraints as equality constraints

Optimisation background

Equality and inequality constraints – Karush-Kuhn-Tucker (KKT) equations

- The system above is composed of the so-called KKT equations.
- We observe that if a constraint $h_i(x) \geq 0$ is inactive *at optimality*, i.e. in x^o, then the corresponding μ_i is zero.
- In the opposite case, the sign of μ_i dictates whether f increases or decreases when entering or exiting the admissible region as dictated by the inequality constraints.
- (*) Assuming that we want to *minimise* (min (cost)) $f(x)$ and the sign in the $h(x)$ equality constraints is $\geq$, requiring that $f(x)$ *increase* when x *enters* the feasibility region for all binding h_i – i.e., that a KKT solution with at least one binding inequality constraint be a candidate *bound minimum* – corresponds to requiring that all nonzero μ_i be *negative* in it. Of course all the other combinations are possible (we may want to maximise $f(x)$ and inequality constraints may have the $\leq$ sign).
- Let us now go for an elementary example.

(*) this can be proven, valid in general this facts!

Example 1

We want to minimize

$$f(x_1,x_2) = x_1^2 + 2x_2^3$$

$$\text{s.t.} \quad x_1 \geq 0, \quad x_2 \geq 0$$

$$(x_1-2)^2 + (x_2-2)^2 = 1$$

→ Interpret Geometrically and write KKT equations.

(sometimes on EXAM KKT equations to write for a given problem!)

KKT)

- FIRST: LAGRANGIAN... (L)

$$f = x_1^2 + 2x_2^3$$

$$g = (x_1-2)^2 + (x_2-2)^2 - 1$$

$$h_1 = x_1$$

$$h_2 = x_2$$

$$L = f + \lambda g + \mu h$$

- $L = x_1^2 + 2x_2^3 + \lambda\left((x_1-2)^2 + (x_2-2)^2 - 1\right) + \mu_1 x_1 + \mu_2 x_2$

- KKT equations:

$$\left.\begin{aligned}
\frac{\partial L}{\partial x_1} = L_{x_1} = 0 &\Rightarrow 2x_1 + 2\lambda(x_1-2) + \mu_1 = 0 \\
L_{x_2} = 0 &\Rightarrow 6x_2^2 + 2\lambda(x_2-2) + \mu_2 = 0 \\
L_\lambda = 0 &\Rightarrow (x_1-2)^2 + (x_2-2)^2 - 1 = 0 \\
\mu_1 \cdot L_{\mu_1} = 0 &\Rightarrow \mu_1 x_1 = 0 \\
\mu_2 \cdot L_{\mu_2} = 0 &\Rightarrow \mu_2 \cdot x_2 = 0
\end{aligned}\right\} \rightarrow$$

solve on MAXIMA:

$[x1=0, x2=2-\sqrt{3}\,\%i, lambda=-\sqrt{3}\,\%i-12, mu1=-4\sqrt{3}\,\%i-48, mu2=0],$

$\left[x1=\sqrt{3}\,\%i+2, x2=0, lambda=\frac{2\sqrt{3}\,\%i-3}{3}, mu1=0, mu2=\frac{8\sqrt{3}\,\%i-12}{3}\right],$

$\left[x1=2-\sqrt{3}\,\%i, x2=0, lambda=-\frac{2\sqrt{3}\,\%i+3}{3}, mu1=0, mu2=-\frac{8\sqrt{3}\,\%i+12}{3}\right]]$

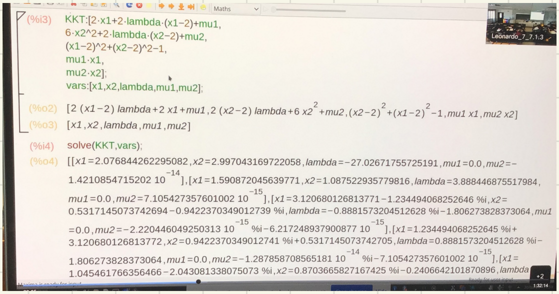
...

numerical solution!

↓

We get 10 solutions

maxima...



GRAPHICALLY

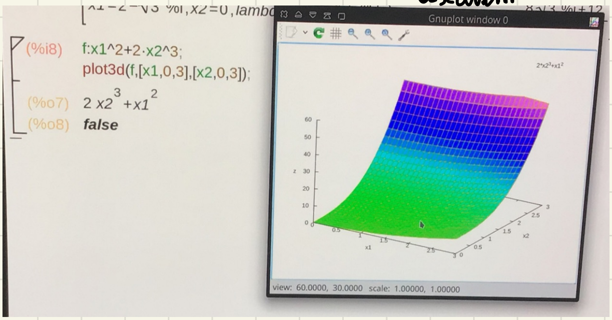
some II m solutions to discard... !

candidates solutions

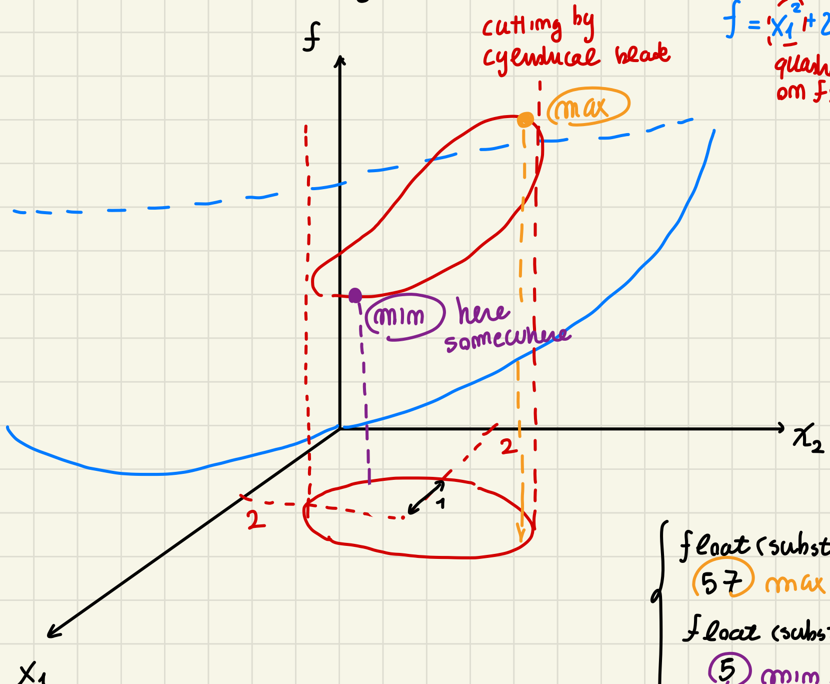
X_1	X_2	
2.07	2.99	← max!
1.59	1.08	← min!

! all the others are complex solutions x_1, x_2 to discard

2 stationary points



INTERPRETATION - Geometry



$f = x_1^2 + 2x_2^3$ — quadratic on $f x_1$ plane; cubic on $f x_2$ plane ↓ grow more on x_2

float(subst([x_1=2.07, x_2=2.99],f);
57 max

float(subst([x_1=1.59, x_2=1.08],f);
5 min!

$(x_1-2)^2+(x_2-2)^2-1=0$ is a circle centered on (2,2) with radius 1

↳ and cutting through → circular blade through the hole

Optimisation background

Equality and inequality constraints – KKT equations – elementary example

$\Downarrow$

(No g(x) here!)

- Minimise $\overbrace{f(x) = x^2}^{f}$ s.t. $x \geq 1$ $(\Rightarrow \overbrace{x-1}^{h_1(x)} \geq 0)$, $x \leq 2$ $(\Rightarrow \overbrace{2-x}^{h_2(x)} \geq 0)$.

  ↑ standard constraint form

- Maxima:

```
f      : x^2;
h1     : x-1;
h2     : 2-x;
L      : f+mu1*h1+mu2*h2;
KKTeqs : [diff(L,x),mu1*diff(L,mu1),mu2*diff(L,mu2)];
solve(KKTeqs,[x,mu1,mu2]);
```

$= 0$

$\mu_2 \dfrac{\partial L}{\partial \mu_2}$

← solve equation!

$\text{diff}(L,x) = \dfrac{\partial L}{\partial x}$

Optimisation background

Equality and inequality constraints – KKT equations – elementary example

- Solutions found: (3 solutions possible)

	x	μ_1	μ_2	$f(x)$	$h_1(x)$	$h_2(x)$	
~~S1~~	0	0	0	0	-1	2	← NO NEG h! constraint violated!
S2	1	-2	0	1	0	1	← Feasible! here h_2 NON BINDING because $\mu_2=0$ while h_1 BIND! $\mu_1 \neq 0$
S3	2	0	4	4	1	0	← Feasible! here h_1 NON BIND, h_2 BIND!

- S1: infeasible.
- S2: feasible, h_2 nonbinding, h_1 binding and entering the feasible region increases f $\Rightarrow$ can be a bound minimum. → entering feasible region increase f (min)!
- S3: feasible, h_1 nonbinding, h_2 binding and entering the feasible region decreases f $\Rightarrow$ cannot be a bound minimum. → f ↓↓ entering feas region MAX

- Hence, the solution is S2.

Optimisation background

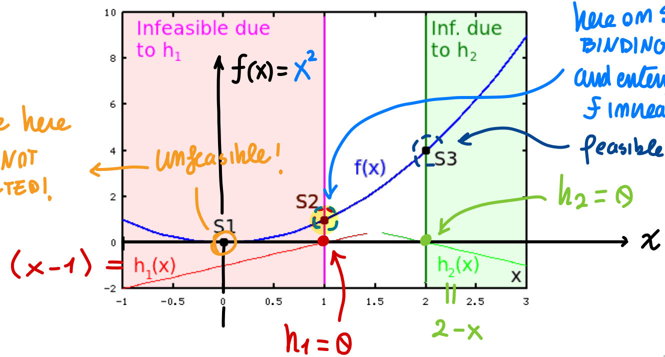
Equality and inequality constraints – KKT equations – elementary example

h=0 identify a point

- This was *really* elementary, so let us have a visual look:

here on S_2 h_1 is BINDING! While h_2 NOT... and entering the region, f increases!

because here $h_1<0$! NOT RESPECTED!



feasible also S_3, on frontier of h_2 and entering feasible region f ↓↓ NO MIN!

- S1: infeasible.
- S2: feasible, h_2 nonbinding, h_1 binding and entering the feasible region increases $f \Rightarrow$ can be a bound minimum.
- S3: feasible, h_1 nonbinding, h_2 binding and entering the feasible region decreases $f \Rightarrow$ cannot be a bound minimum.

Operationally very simple to compute! DISCARD solution and
choose the Best

Math behind defined...

Global cost minimisation

respect the EXAMPLE on slide 206 → dividing boundaries

respect the 2 gen case

The previous two-generator case revisited with the KKT equations

- We are of course looking at the G1+G2 combination, thus the problem is minimise $f(P_{g1},P_{g2}) = c_1(P_{g1}) + c_2(P_{g2})$
  s.t. $P_{g1}+P_{g2} = \widehat{P}_e$, $P_{g1,min} \le P_{g1} \le P_{g1,max}$, $P_{g2,min} \le P_{g2} \le P_{g2,max}$.
- Maxima (preliminaries):

```
Pg1min : 20;
Pg1max : 100;
Pgo1   : 80;
ro1    : 3;
rml1   : 5;

Pg2min : 5;
Pg2max : 50;
Pgo2   : 35;
ro2    : 3.5;
rml2   : 8;

k11    : (rml1*Pgo1^2-ro1*Pg1min*(2*Pgo1-Pg1min))/(Pgo1-Pg1min)^2;
k21    : 2*Pgo1*(ro1-rml1)/(Pgo1-Pg1min)^2;
k31    : (rml1-ro1)/(Pgo1-Pg1min)^2;
k12    : (rml2*Pgo2^2-ro2*Pg2min*(2*Pgo2-Pg2min))/(Pgo2-Pg2min)^2;
k22    : 2*Pgo2*(ro2-rml2)/(Pgo2-Pg2min)^2;
k32    : (rml2-ro2)/(Pgo2-Pg2min)^2;

c1     : k11*Pg1+k21*Pg1^2+k31*Pg1^3;
c2     : k12*Pg2+k22*Pg2^2+k32*Pg2^3;
f      : c1+c2;
```

Global cost minimisation

The previous two-generator case revisited with the KKT equations

- Maxima (we consider the two cases with $\widehat{P}_e$ equal to 70 and 100):

```
/* Set Pe to the desired value, 70 or 100 for the two cases shown */
Pe      : 70;

g       : Pg1+Pg2-Pe;
h1      : Pg1-Pg1min;
h2      : Pg1max-Pg1;
h3      : Pg2-Pg2min;
h4      : Pg2max-Pg2;

L       : f+lambda*g+mu1*h1+mu2*h2+mu3*h3+mu4*h4;
KKTeqs  : [diff(L,Pg1),diff(L,Pg2),
           diff(L,lambda),
           mu1*diff(L,mu1),mu2*diff(L,mu2),
           mu3*diff(L,mu3),mu4*diff(L,mu4)];
S       :  solve(KKTeqs,[Pg1,Pg2,lambda,mu1,mu2,mu3,mu4]);

for i:1 thru length(%rnum_list) do S:subst(t[i],%rnum_list[i],S);
float(S);
fvals   : float(makelist(subst(S[i],f),i,1,length(S)));
```

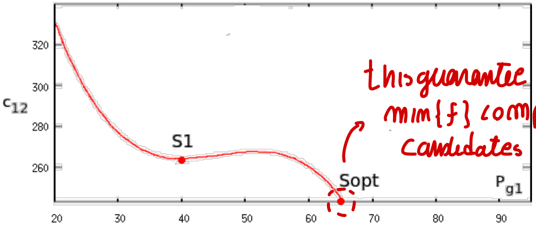
Global cost minimisation

The previous two-generator case revisited with the KKT equations

solving for $\hat{P}_e = 70$

- Results for $\widehat{P}_e = 70$:

P_{g1}	P_{g2}	λ	μ_1	μ_2	μ_3	μ_4	f	
20.00	50.00	any	$-\lambda-\frac{11}{3}$	0.00	0.00	$\lambda+\frac{97}{8}$	331.25	left extremum
51.58	18.42	-1.82	0.00	0.00	0.00	0.00	267.67	local max
40.09	29.91	-2.11	0.00	0.00	0.00	0.00	264.30	S1
100.00	-30.00	-44.12	0.00	-38.68	0.00	0.00	-416.52	infeasible
65.00	5.00	-2.04	0.00	0.00	-4.46	0.00	243.13	Sopt



this guarantee the min{f} computing all candidates!

- Note that the left extremum cannot be a bound minimum, while the right one actually is.

Global cost minimisation

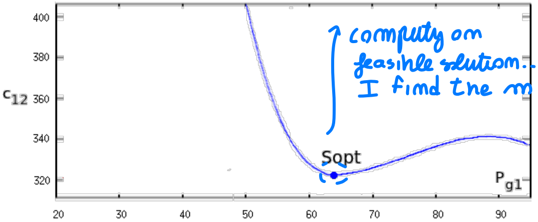
The previous two-generator case revisited with the KKT equations

computation done
with our procedure
and solve with software
(for big network → dedicated SW)

- Results for $\widehat{P}_e = 100$:

P_{g1}	P_{g2}	λ	μ_1	μ_2	μ_3	μ_4	f	
89.75	10.25	-4.02	0.00	0.00	0.00	0.00	341.26	local max
69.42	30.58	-2.25	0.00	0.00	0.00	0.00	322.59	Sopt
20.00	80.00	-49.63	45.95	0.00	0.00	0.00	1190.00	infeasible
100.00	0.00	-9.63	0.00	-4.18	0.00	0.00	322.22	infeasible
95.00	5.00	-4.71	0.00	0.00	-1.79	0.00	336.88	right extremum
50.00	50.00	-1.83	0.00	0.00	0.00	10.29	406.25	left extremum

check the best
possible solution!
using our procedure
we find possible candidates
and compare function
values!



- Note that the left extremum cannot be a bound minimum, while the right one could but is not.
- Recall that the KKT equations not only are necessary conditions, but just provide *candidate optima*: always check the solutions!

RECAP ⇝ for electrical syst:

- we can set up primary and secondary control given responce speed and max freq error requirements
- compute tertiary control balances by using KKT equations
- single objective (min cost)
  OR I can do this with partial objectives
  ↳ only min cost of some gen while other @ default

↓

in this case partial obj: problem structure is the same

it changes the problem structure with another viewpoint: totally sefish utility viewpoint

↓

respect Generators or Utilizers

Preliminaries

The generator viewpoint

there are no means of influencing the price (energy price fixed!)

- Assume you run a generator (or pool of) and that you are a $\underbrace{\text{price taker}}$.
- Let π_σ – say [€/MWh] – be the energy selling price.
- Let $c(P_g)$ [€/h] be your cost rate as function of your generated power P_g [MW].
  over time "rate"
- To maximise your revenue (rate) you state the problem
  (derivative of revenue)

[MWh] = [J]

what you get!

[€/J] * [J/s] = [€/s]

$$\boxed{\max_{P_g} \left(\pi_\sigma P_g - c(P_g)\right)}$$

what you spend!

cost rate!

and readily get the production you desire by solving for P_g the equation

$$\boxed{\frac{dc(P_g)}{dP_g} = \pi_\sigma .}$$

so overall derivative of

$$\frac{d}{dP_g}\left(\pi_\sigma P_g - c(P_g)\right) = 0$$

Preliminaries → Graphical view:

The generator viewpoint – example

- Maxima:

```
c   : 6*Pg-0.01*Pg^2+0.002*Pg^3;
sol : solve(diff(c,Pg)=psigma,Pg);
Pgo : rhs(sol[2]);
      float(subst(psigma=50,Pgo));
      plot2d([diff(c,Pg),50],[Pg,20,100]);
```

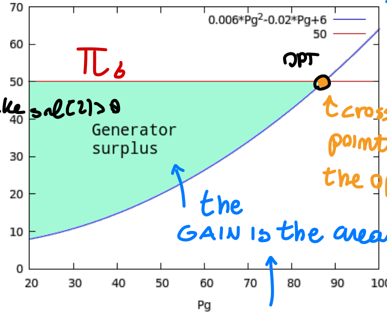
(invented)

→ sol[1]<0

→ Just take the numb. ↲ I take sol[2]>0

right hand ↗ side of the solution, so the value of (x=-1)

- The obtained optimum P_g is 87.32.
- Let us now plot $dc(P_g)/dP_g$ versus P_g and check the crossing with π_σ.
- NOTE: should the optimum P_g be above the generator capacity, the desired generation and the surplus would be bound to that maximum capacity.



$\frac{dC(P_g)}{dP_g}$

π_σ

OPT

↑ cross point is the opt!

↑ the GAIN is the area!

↑ became is the Gen. service

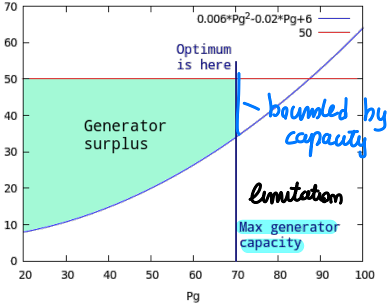
earn more and spend more ←

above OPT. you spend more than how much you gain (more cost increase!)

Preliminaries

The generator viewpoint – peculiar cases

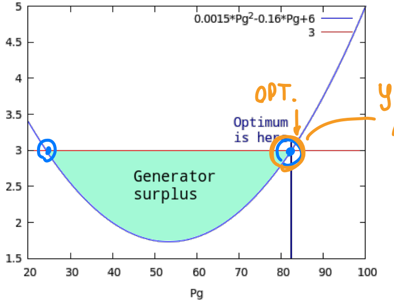
NON monotonic derivative!



bounded by capacity

limitation

Bound optimum.



OPT.

you want to maximize the area (surplus) so you take this as OPT

High-inflection cost rate curve.

in some cases...

Preliminaries

The utiliser viewpoint

- Assume now you are an energy utiliser — and again, a price taker.
- Let π_σ – say again [€/MWh] – be the energy selling price.
- Assume your utility function $U(P_t)$ – [€/h], where P_t is the taken power – to be "well-behaved" in economic jargon, that is, monotonically increasing and "(quasi-)concave"; a typical form can be cubic without constant term and with negative second derivative in the range of interest. ~> for example $300P_t - 3.6P_t^2 + 0.015P_t^3$
- To maximise your utility (rate) you state the problem

"quasi-concave because little concavity change

$$\boxed{\max_{P_t}\left(U(P_t) - \pi_\sigma P_t\right)}$$

maximize the earnings → max diff of utility for consumption and how much I spent

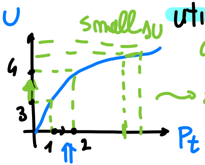
and get your optimal consumption by solving for P_t the equation

↓ monotonically increase but on that range not soo much

$$\frac{dU(P_t)}{dP_t} = \pi_\sigma.$$

Preliminaries

The utiliser viewpoint – example



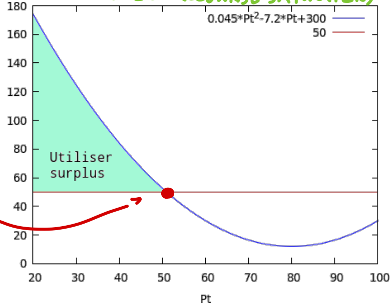
utility curve → increment in consumption cause U ↑ (utility ≃ usefull effect given by energy)

→ for high consumption, an increase on consump Pt cause less ΔU (almost SATURATED) → "QUASI" CONCAVE

- Maxima:

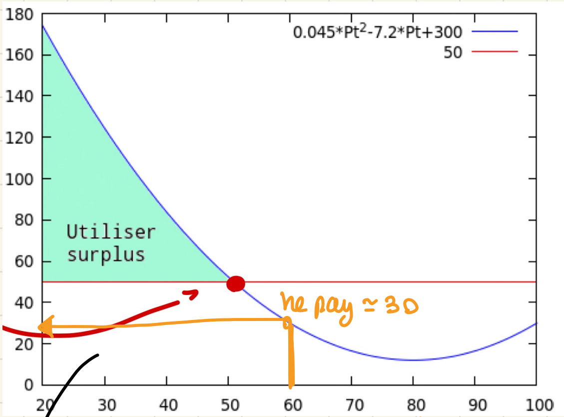
```
U   : 300*Pt-3.6*Pt^2+0.015*Pt^3;
sol : solve(diff(U,Pt)=psigma,Pt);
Pto : rhs(sol[1]);
      float(subst(psigma=50,Ptopt));
      plot2d([diff(U,Pt),50],[Pt,20,100]);
```

↑ plot ∂U/∂Pt →



- The obtained optimum P_t is 50.94.
- Let us now plot $dU(P_t)/dP_t$ versus P_t and check the crossing with π_σ.
- NOTE: should the optimum P_g be above the utilser capacity, the desired utilisation and the surplus would be bound to that maximum capacity.

↓ IF you consume more than this Pt choice → the additional utility is NOT compensated by how much you spent!



offer to produce 60

Usefull to make offers

↓

Knowing the price 50

next period I can offer 24 .. less...

OR selling price 50

↳ next period I can offer the best...

{ obtained optimum P_g }

↓

respect surplus

IF someome propose a price ⇝ I can propose a consumption

OR viceversa...

IF I come the producer. gen

offering a production of 70 ⇒ IF you pay 20 to be convenient!

↓

dually for Gen. POV

Relating the two viewpoints

→ this allows you to make offers! IF I know a certain price, I can lock
BUT on next period
(complicated bidding mechanism)

- Given a time slot in the future, generators and utilisers can bid for a certain amount of produced/consumed energy at a certain price in that slot. In a nutshell, this is how energy is traded. → fixing the price on that slot!
- The matter is extremely complicated, in fact, involving auctions at various space and time scales.
- However this is a course on automation, hence we are not interested in the details of the bidding process – which is more a matter of economic management – but rather on how this process reflects on the control layers for the network.
- As such, we are now going for a wrap-up from our perspective.

Ingredients of the optimisation problem

- Every generator has a cost model providing the cost rate [€/h] as a function of the produced power [MW].
- Such cost (rate) functions are well represented, in general, by polynomials up to cubic.
- Utilisers of a certain size have – more or less explicitly – a utility function relating some KPI (for example an income rate) to the consumed power.
- Apparently the last idea makes little sense for a single house...
- ...but it could, no matter the KPI, e.g. for a district or a municipality.
- Such utility (rate) functions are, or at least in general are assumed to be, "well-behaved" (monotonic, increasing, concave).

Actors of the optimisation problem

ACTORS { Generators (large), prosumers, TSO

- In the traditional *scenario*, generators control the network.
- Recently, utilisers can participate
  - by reducing (shaving) the load,
  - by shifting their utilisation in time, often combined to shaving,
  - and with similar manoeuvres.
- In the presence of distributed generation, a compound of utilisers can also play the role of a generator of relevant size: the term "prosumer" (producer + consumer) was introduced to indicate this double utiliser role (producer and consumer).
- The third class of actors comprehends national/regional Grid Managers or TSOs (Transmission System Operators) with the role of
  - maintaining continuous production/consumption balance (therefore controlling frequency), ensure that @any time there is a way to increase production
  - ensuring the "rotating reserve" provision (some words on this later),
  - selecting the optimal generation distribution for each trading period,
  - consequently instructing generators about how much they have to produce, trading period per trading period.

"rotating mass"

The optimisation problem

- We know how a TSO can optimise a cost function network-wide (we did this for total cost via the KKT equations). → solved by dedicated SOFTWARE
- We shall see later on (talking about load flow) that additional constraints cen be introduced to not overload transmission lines — but the *concept* does not change.
  (still optimiz. problem)
- We know how a generator/utiliser can determine its optimal operating conditions in a selfish manner. → {like to produce/consume some given the price! like to have a certain price given consume/production!}
- Determining the TSO decision is an extremely complicated procedure (and in fact, not strictly "automation"). ↳ determin price and devices
- We provide a high-level overview, and then come back to control.

The optimisation problem

in a simplified world for clarity

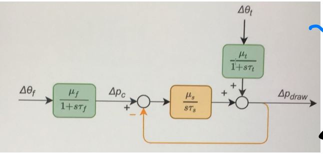
- The TSO – or some other entity, either part of the TSO or coordinated, called in general the "electricity market" or "power exchange" or PX – comes to setting the energy price based on the bids to buy and the offers to sell (this is called short-term trading, we do not talk here about long-term operations). → economic aspect... financial stuff
- The TSO operates on a future trading period, knowing
  - a forecast of the demand (e.g., from historical data, meteo, and so forth),
  - the trading results for the period under question,
  - the generator characteristics (e.g., base load or available for slow or fast variations),
  - possible global requirements (cost, pollution, use of renewables, and so on).
- The TSO detremines for each generator
  - the production (constant) for the period, which is tertiary control,
  - and possibly the amount of additional production (rotating reserve) that the generator has to guarantee, via primary/secondary control, if needed to keep frequency at the set point; there is in general some remuneration for providing reserve, we omit details.

The resulting overall scheme

- Tertiary control results are sent to the generator controls as bias values, resulting in the scheme below (where we show that the secondary control distribution can be more complex than just a β_i, as we write for practical reasons in our exercises).

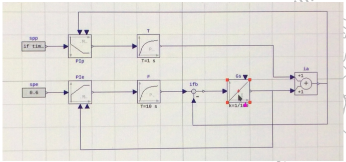
"Modelica": Elecsys_

talking about...



the same

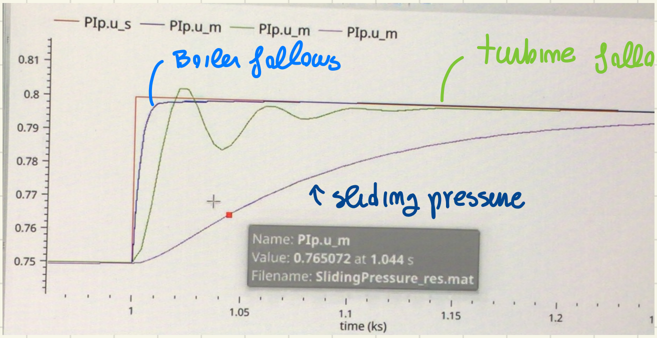
Boiler follows



+ other models on > Elect Sys _ generator models

⇓

running
system with same input signal (possible responce)



Boiler follows

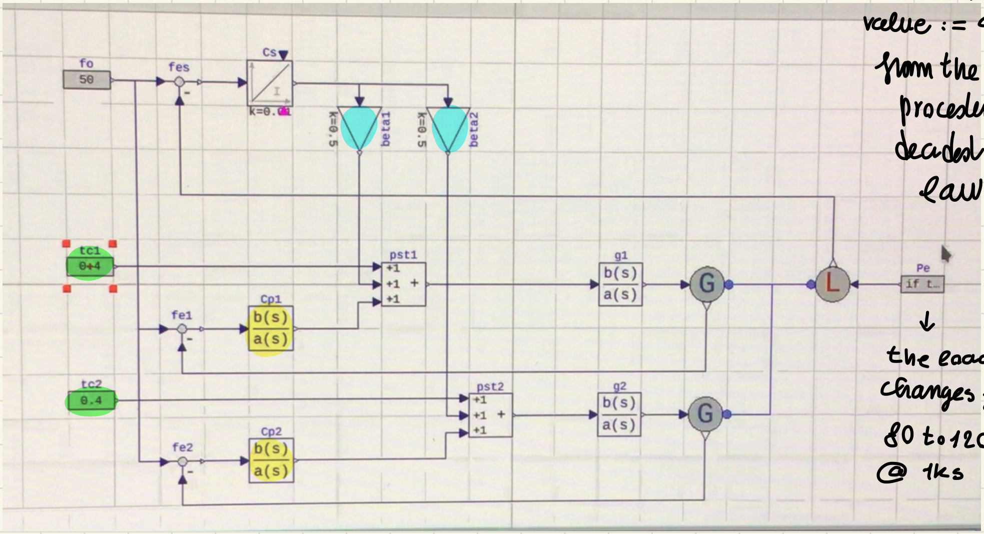
turbine follows (oscillate to be fast)

↑ sliding pressure

\+ > ElecSys - power - frequency _ control

\+ > Elec Sys - case - studies
Primary - secondary - tertiary

primary
secondary
tertiary → tertiary is a bias added
↳ as constant value := coming from the optimization procedure, not decided by control law scheme!



↓
the load changes from 80 to 120 @ 1ks

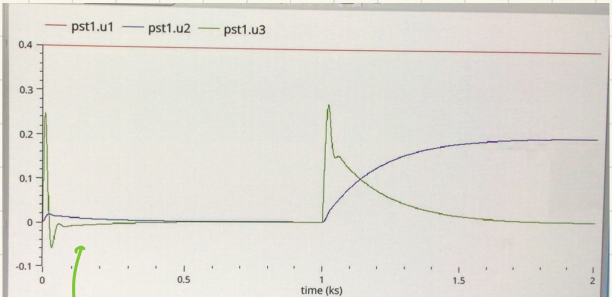
-- running this...

↓

on summation mode... total request from

- p (u_3)
- s (u_2)
- t (u_1)

control



primary provide the transient contribution!