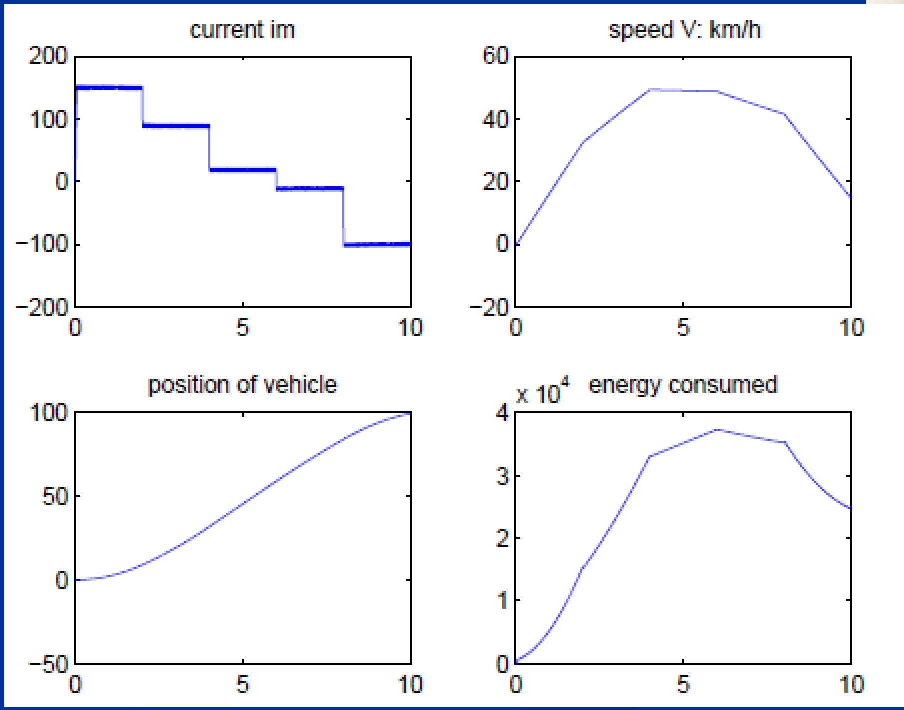
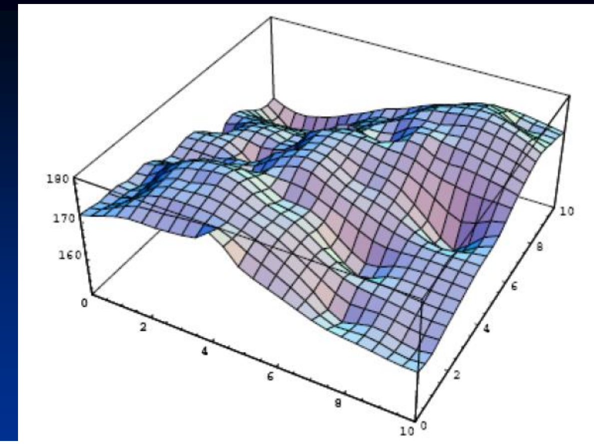
# On computational Dynamic Programming for minimizing energy in an electric vehicle

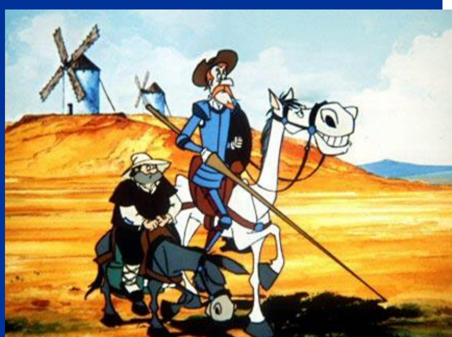


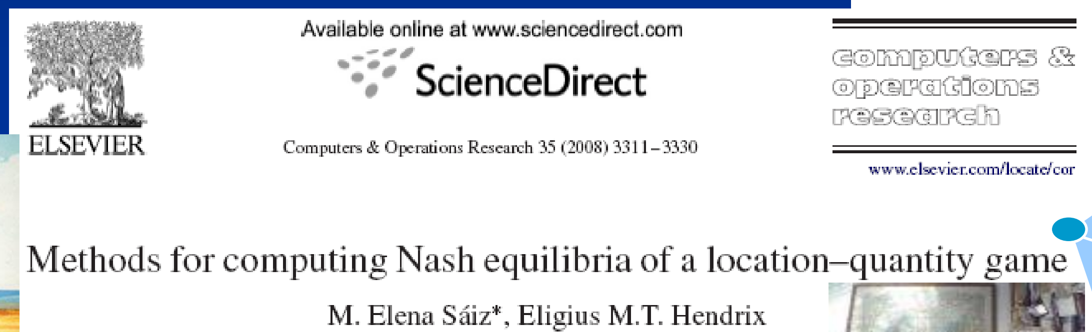

Inmaculada Garcia
Eligius M.T. Hendrix
Europe
18 slides

31 January 2019

# Location before coming to Spain



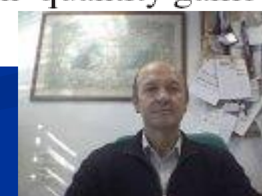



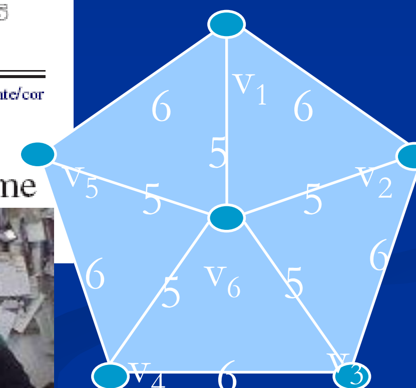


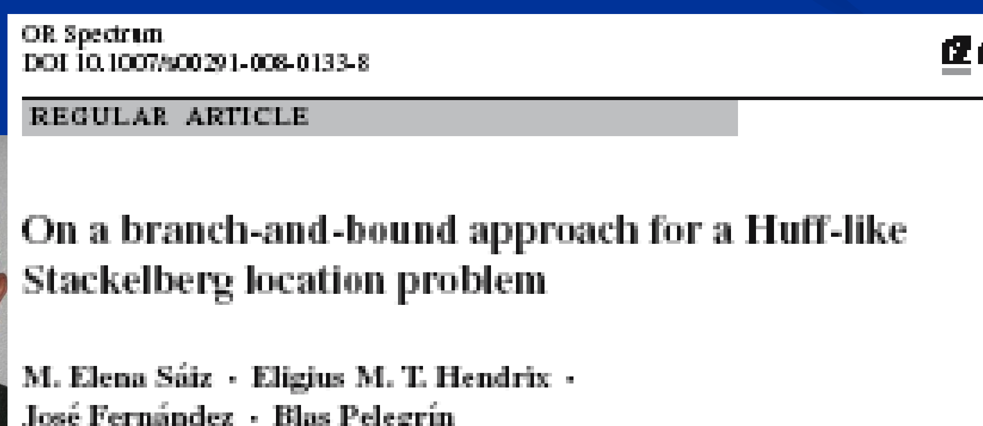
ELSEVIER
Available online at www.sciencedirect.com
ScienceDirect
Computers & Operations Research 35 (2008) 3311–3330
computers & operations research
www.elsevier.com/locate/cor


Methods for computing Nash equilibria of a location–quantity game


M. Elena Sáiz*, Eligius M.T. Hendrix



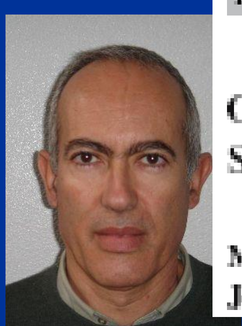



OR Spectrum
DOI 10.1007/s00291-008-0133-8
Open Access


REGULAR ARTICLE

On a branch-and-bound approach for a Huff-like Stackelberg location problem


M. Elena Sáiz · Eligius M. T. Hendrix ·
José Fernández · Blas Pelegrín


# Lets go South, 2008

EWGLA_08 Elche
SEIO_08 Murcia


# Do you want to be a member of RED-LOCA?

ELSEVIER journal homepage: www.elsevier.com/locate/ejor

Continuous Optimization

## Locating a competitive facility in the plane with a robustness criterion

R. Blanquero[a,*], E. Carrizosa[a], E.M.T. Hendrix[b]

[a] Facultad de Matemáticas, Universidad de Sevilla, Tarfia s/n, 41012 Sevilla, Spain
[b] Computer Architecture, Universidad de Málaga and Operations Research and Logistics, Wageningen University, The Netherlands

.com/locate/ejor

## On Nash equilibria of a competitive location-design problem

M. Elena Sáiz[a,*], Eligius M.T. Hendrix[b,1], Blas Pelegrín[c]

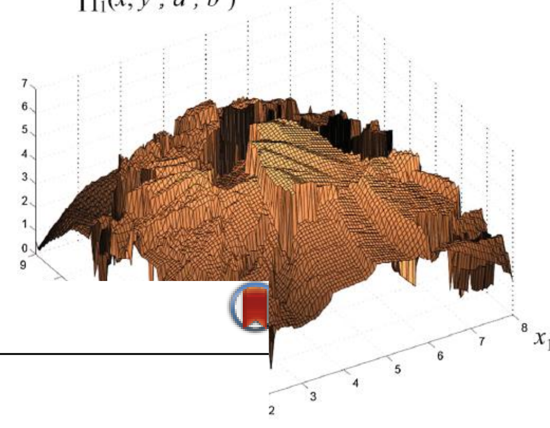
[a] Methoden, Radboud Universiteit Nijmegen, The Netherlands
[b] Computer Architecture, Universidad de Málaga and Operations Research and Logistics, Wageningen Universiteit, The Netherlands
[c] Dpt.Estadística e Investigación Operativa, Universidad de Murcia, Spain

.com/locate/ejor

## Recent insights in Huff-like competitive facility location and design

José Fernández[a], Eligius M.T. Hendrix[b,*]

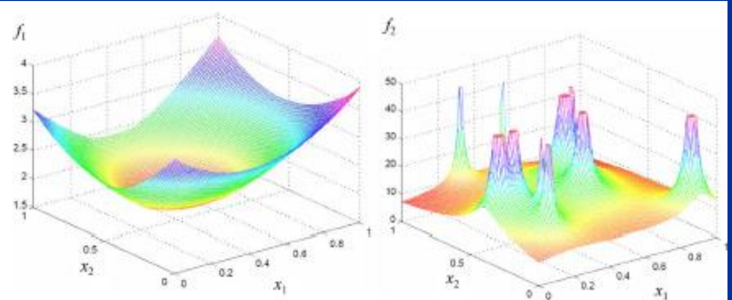
[a] Dpt. Statistics and Operations Research, University of Murcia, Spain
[b] Computer Architecture, University of Málaga and Operations Research and Logistics, Wageningen University, Ne




Comput Optim Appl (2015) 61:205–217
DOI 10.1007/s10589-014-9709-1

## On heuristic bi-criterion methods for semi-obnoxious facility location

P. M. Ortigosa · E. M. T. Hendrix · J. L. Redondo




Graph minsum (Weber) function $f_1$   Graph obnoxious function $f_2$

Ann Oper Res (2016) 246:19–30
DOI 10.1007/s10479-015-1793-9

## On competition in a Stackelberg location-design model with deterministic supplier choice

Eligius M. T. Hendrix

# Do you want to be a member of RED-LOCA?

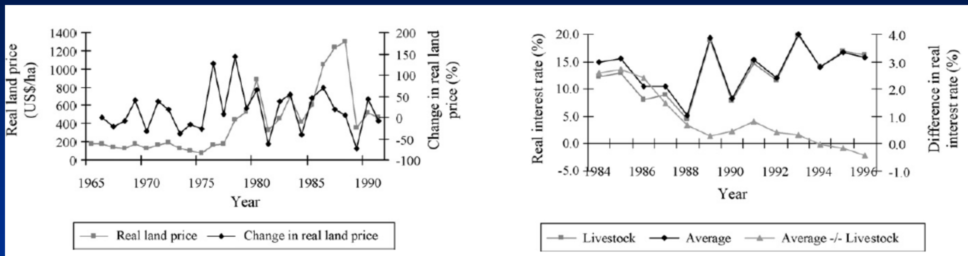
VIII Workshop on Locational Analysis and Related Problems 2017 1

## On location and vessel fleet composition for offshore wind farm maintenance*

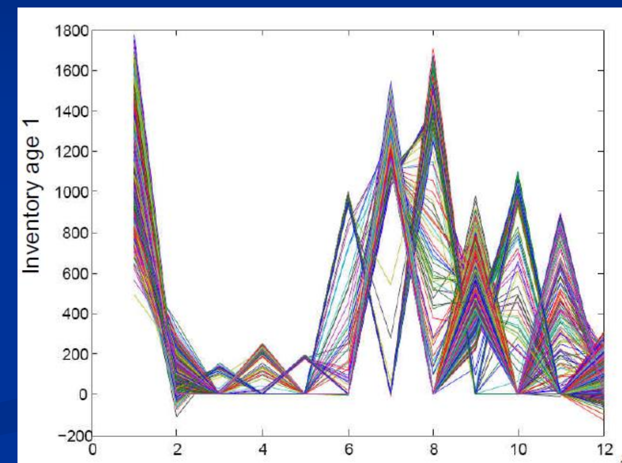
A.G. Alcoba[1], E.M.T. Hendrix[1], G. Ortega[2] D. Haugland[3], E.E. Halvorsen-Weare [4]

Topics seem to change subject

# TSP in dynamic programming is biutiful

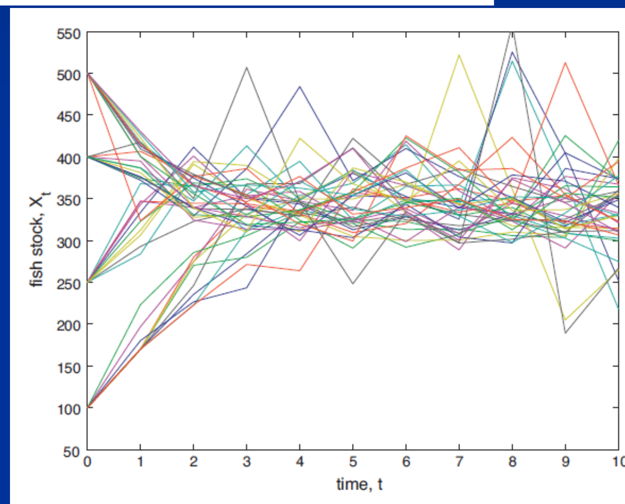



From 2012 SDP in Inventory control

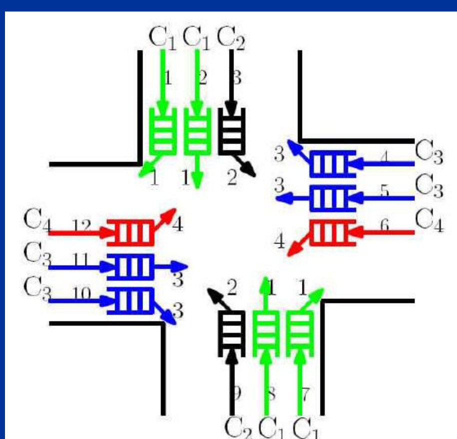


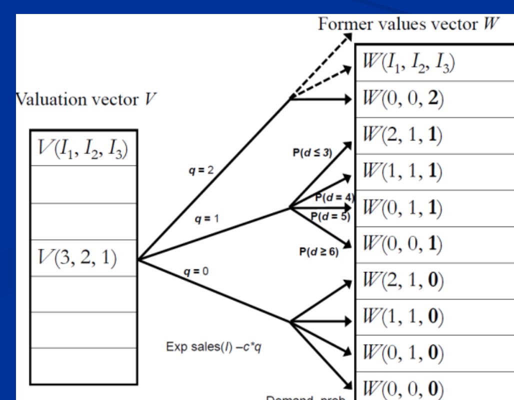

2010-2013

- Deforestation
- Fishery quota
- Traffic control
- Climate change




Non-stationary, finite horizon
Stationary, Markov



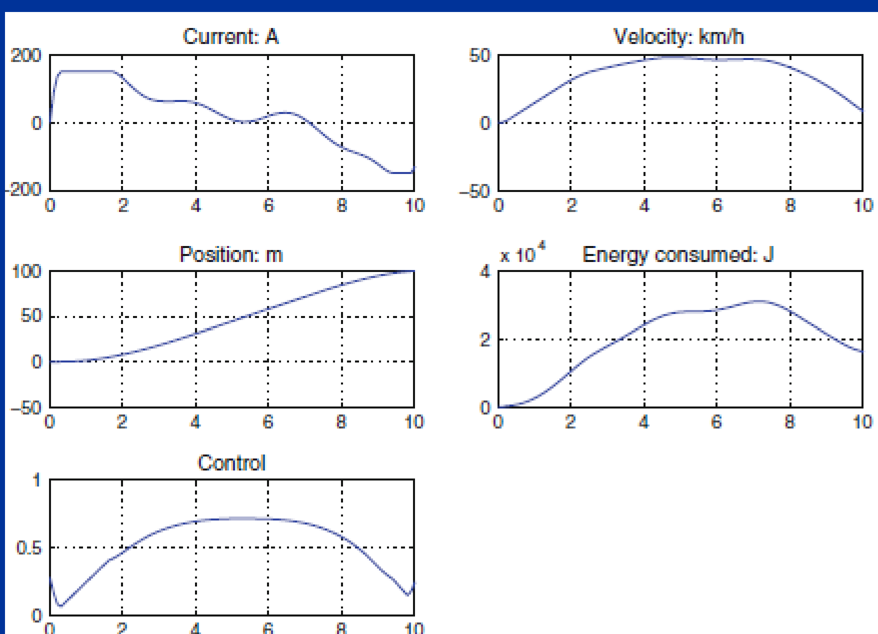



# Can I talk about electric cars, RED-LOCA?

Proceedings of TOGO 2010, pp. 85 – 88.

**Toward Global Minimum Solutions for the Problem of the Energy Consumption of an Electrical Vehicle**

Abdelkader Merakeb[1,2] and Frédéric Messine[2]


4OR-Q J Oper Res (2014) 12:261–283
DOI 10.1007/s10288-013-0247-y

RESEARCH PAPER

**A Branch and Bound algorithm for minimizing the energy consumption of an electrical vehicle**

Abdelkader Merakeb · Frédéric Messine ·
Mohamed Aidène

Why to use B&B?

# Can I read engineerish?


$$
\begin{cases}
\min_{i_m(t),\Omega(t),pos(t),u(t)} \quad E(t_f, i_m, u) \\
s.t. \\
\quad \frac{di_m(t)}{dt} = \frac{u(t)V_{alim} - R_m i_m(t) - K_m \Omega(t)}{L_m} \\
\quad \frac{d\Omega(t)}{dt} = \frac{1}{J}\left(K_m i_m(t) - \frac{r}{K_r}\left(MgK_f + \frac{1}{2}\rho S C_x \left(\frac{\Omega(t) r}{K_r}\right)^2\right)\right) \\
\quad \frac{dpos(t)}{dt} = \frac{\Omega(t) r}{K_r} \\
\quad |i_m(t)| \leq 150 \\
\quad u(t) \in \{-1, \ +1\} \\
\quad (i_m(0), \ \Omega(0), \ pos(0)) = (i_m^0, \ \Omega^0, \ pos^0) \in \mathbb{R}^3 \\
\quad (i_m(t_f), \ \Omega(t_f), \ pos(t_f)) \in \mathcal{T} \subseteq \mathbb{R}^3
\end{cases}
$$

an electrical solar car: $K_r = 10$, the coefficient of reduction; $\rho = 1.293$ kg/m$^3$, the air density; $C_x = 0.4$, the aerodynamic coefficient; $S = 2$ m$^2$, the area in the front of the vehicle; $r = 0.33$ m, the radius of the wheel; $K_f = 0.03$, the constant representing the friction of the wheels on the road; $K_m = 0.27$, the coefficient of the motor torque; $R_m = 0.03$ Ohms, the inductor resistance; $L_m = 0.05$, the inductance of the rotor; $M = 250$ kg, the mass; $g = 9.81$, the gravity constant; $J = M \times r^2 / K_r^2$; $V_{alim} = 150$ V, the battery voltage; $R_{bat} = 0.05$ Ohms, the resistance of the battery.

¿Que??

# Translated to Optimizish?

Indices

| | |
|---|---|
| $t$ | Moment in time with $\delta = 0.1$ second slots, $t = 0, \ldots, T$ |

Parameters

| | |
|---|---|
| $H$ | Final control horizon in seconds |
| $\delta$ | time discretization slot, $\delta = 0.1$ |
| $T$ | Number of periods in the horizon $T = \frac{H}{\delta}$ |
| $P$ | Target position to be reached in control horizon |
| $R$ | Radius of the wheels in m |
| $B$ | Resistance of the battery, $B = .05$ Ohm |
| $S$ | Voltage of power supply, $S = 150$ volts |
| $Tr$ | Transmission coefficient motor to wheels, $Tr = 10$ |
| $C$ | resistance depending on air density, surface car and aerodynamics, $C = .517$ |
| $L$ | Inductance rotor, $L = .05$ |
| $I$ | Inductor resistance, $I = .03$ Ohm |
| $Q$ | Coefficient motor torque, $Q = .27$ |
| $M$ | Mass vehicle, $M = 250$ kg |
| $G$ | Gravity constant, $G = 9.81$ |
| $F$ | Friction coefficient of the wheels, $F = .03$ |


Variables

| | |
|---|---|
| $i_t \in [-150, 150]$ | Induction of the engine |
| $\omega_t$ | radial speed in radius per second. This translates to the velocity $v_t = \frac{R}{Tr}\omega_t$ in meter per second |
| $p_t \in [0, P]$ | position of the vehicle |
| $u_t \in \{-1, 1\}$ | Control, switch. One can switch very frequently, such that this variable can also be considered continuous. We will make use of that to limit its value such that $i_t \in [-150, 150]$ |

# And stare at the model?

The objective is given by

$$E = \delta \sum_{t=0}^{T-1} Su_t i_t + Bu_t^2 i_t^2. \tag{1.1}$$

The dynamics is given by difference equations taking the time step size $\delta$ into account. Position:

$$p_t = p_{t-1} + \delta v_t. \tag{1.2}$$

Induction:

$$i_t = i_{t-1} + \delta \frac{Su_t - Ii_{t-1} - Q\omega_{t-1}}{L}. \tag{1.3}$$

The dynnamics of the radial speed is given by

$$\omega_t = \omega_{t-1} + \delta \frac{Tr}{R} \left( \frac{QTr}{RM} i_{t-1} - GF - \frac{C}{M} v_{t-1}^2 \right). \tag{1.4}$$

Variables

| | |
|---|---|
| $i_t \in [-150, 150]$ | Induction of the engine |
| $\omega_t$ | radial speed in radius per second. This translates to the velocity $v_t = \frac{R}{Tr}\omega_t$ in meter per second |
| $p_t \in [0, P]$ | position of the vehicle |
| $u_t \in \{-1, 1\}$ | Control, switch. One can switch very frequently, such that this variable can also be considered continuous. We will make use of that to limit its value such that $i_t \in [-150, 150]$ |

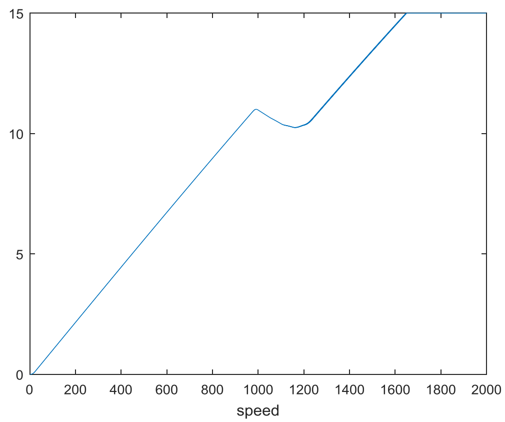
DP view, find control rule $u_t(i, \omega, p)$ obviously

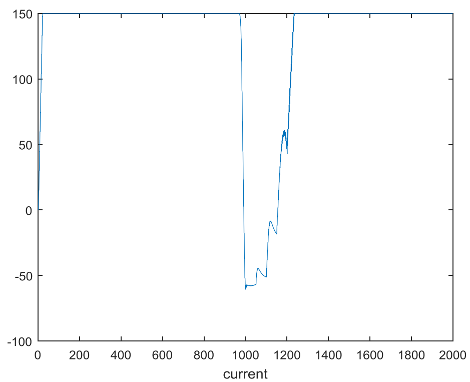
# And …….solve…. or not

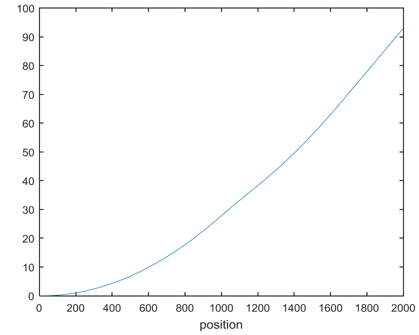
DP view:

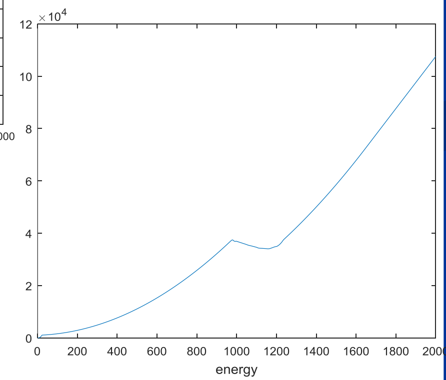
- Build simulation model, small time steps
- Bound state space
- Grids on state space and use interpolation (continuous variables)
- Run Bellman recursion backwards, with penalties on bad states
- Harvest a table u(t,i,ω,p) control rule $u_t(i,\omega,p)$
- Simulate applying interpolation
- Some pain in the discretisation of the time. Simulating forward different from step sizes control.

First implementation horrible

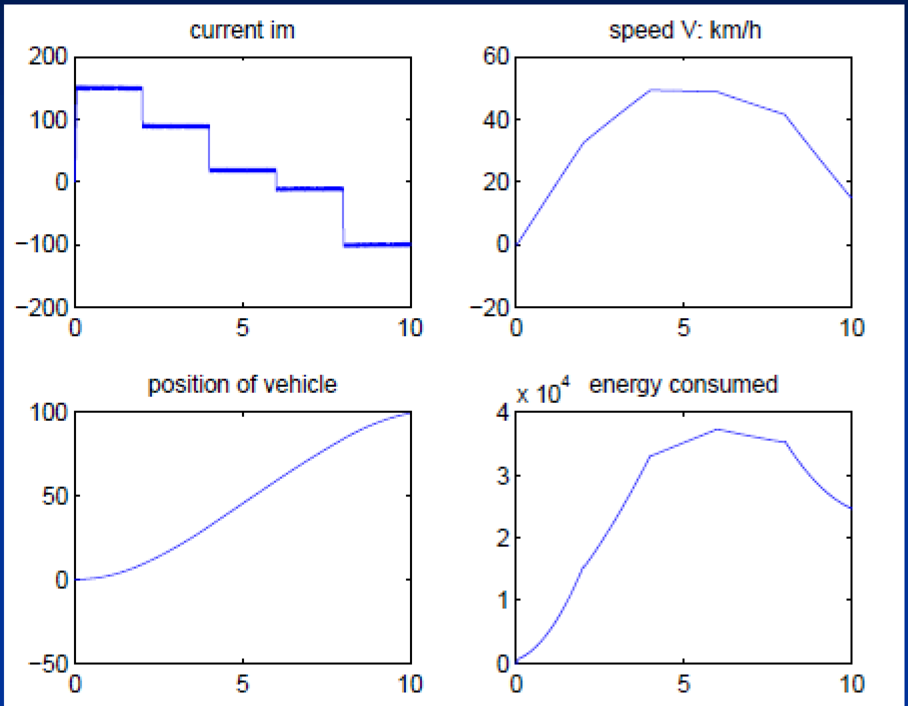


# Three weeks ago
# Can I repeat the earlier exercise?





Ana Rocha,
Universidade do Minho


Such a thing should have come out.

Let us try, nonlinear optimisation on $u_t$ ϵ [-1,1] and limiting control to

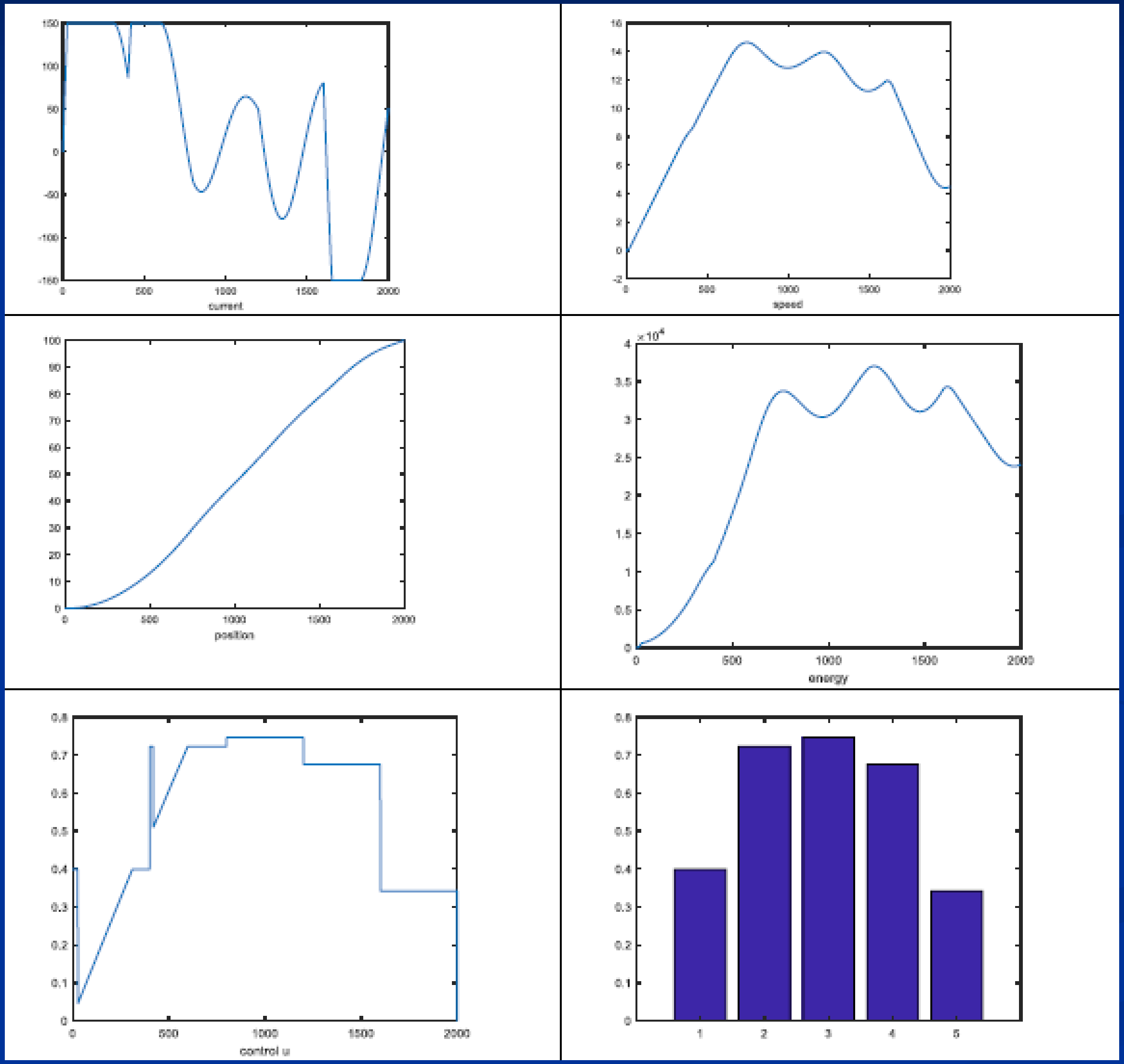
$$\Delta_t = \frac{150L + (\delta I - L)i_{t-1} + \delta Q\omega_{t-1}}{\delta S}$$

values of the control to

$$u_t \in \{min\{-1, \text{sgn}(\Delta_t)\min\{|\Delta_t|, 1\}, \max\{1, \text{sgn}(\Delta_t)\min\{|\Delta_t|, 1\}\}$$

Within the simulator with small time steps of δ=.002

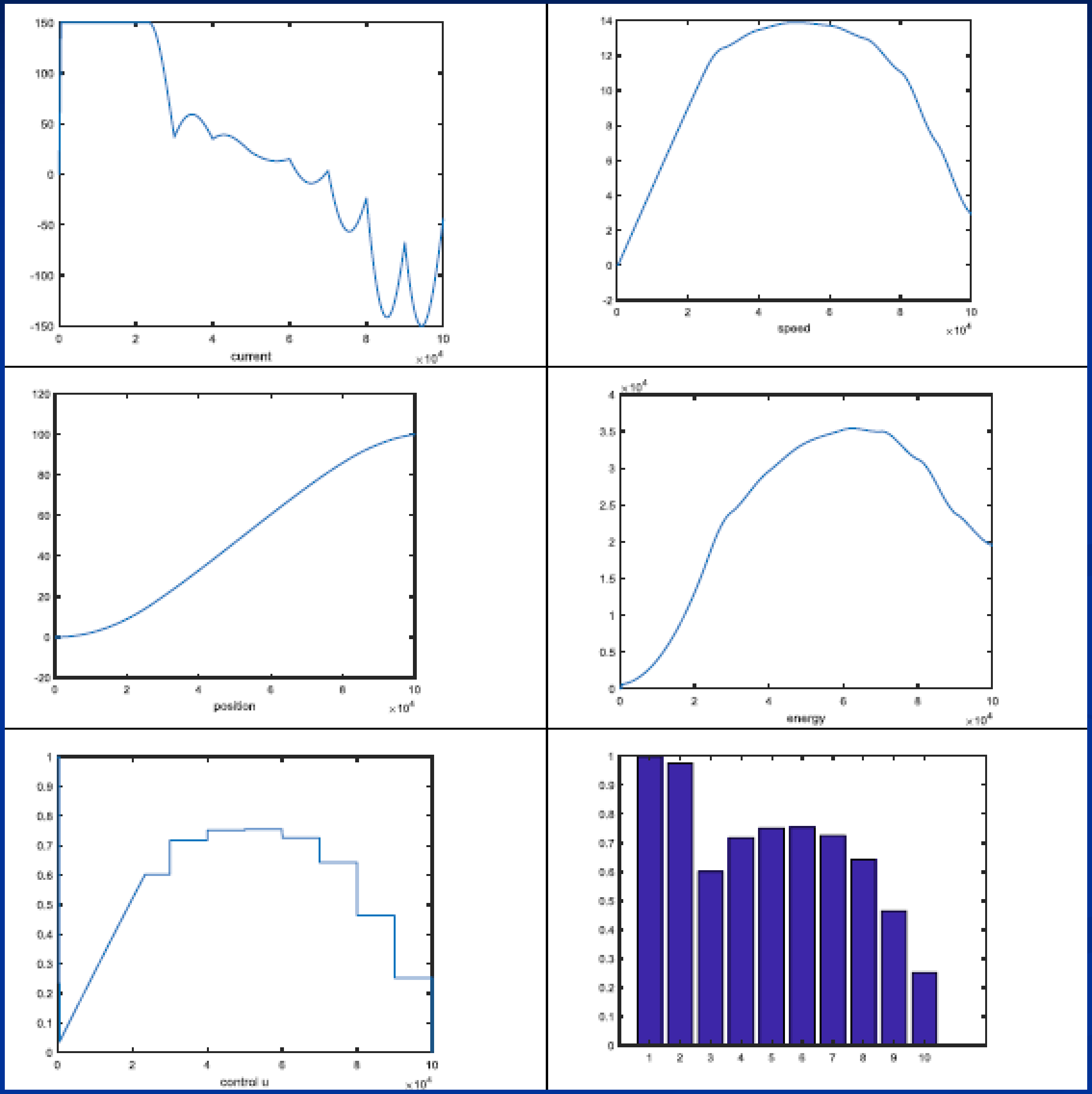
# Three weeks ago
# Can I repeat the earlier exercise?




5 values for *u*
Sim-based optimisation
fmincon

Uf, makes more sense
Energy 24191 joules

# Three weeks ago
# Can I repeat the earlier exercise?

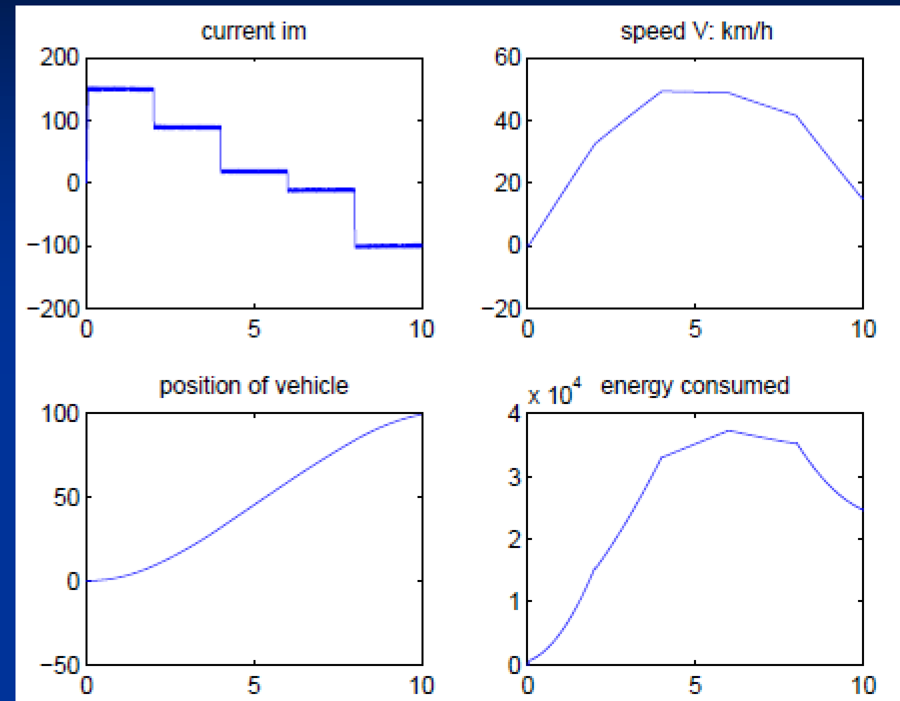


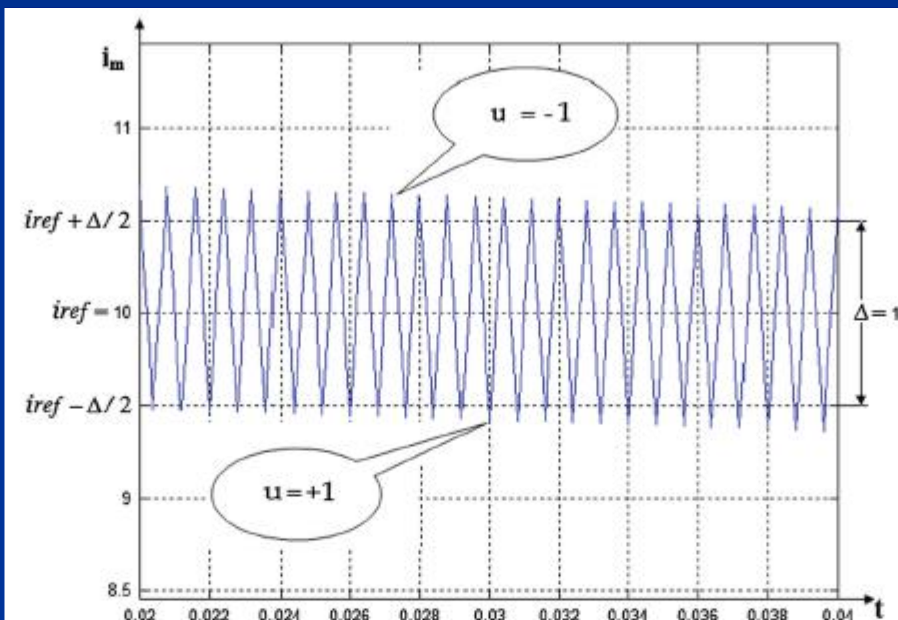

10 values for *u*
Sim-based optimisation
fmincon

Energy 19496 joules

# Can I repeat the earlier exercise?
# Not exactly, what they had been doing is

Optimise reference values for current *i*



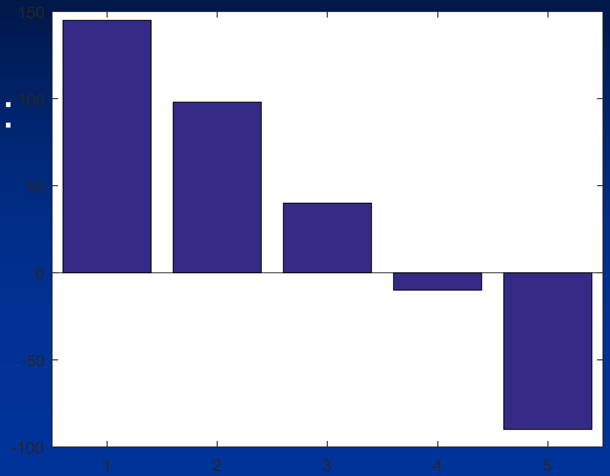



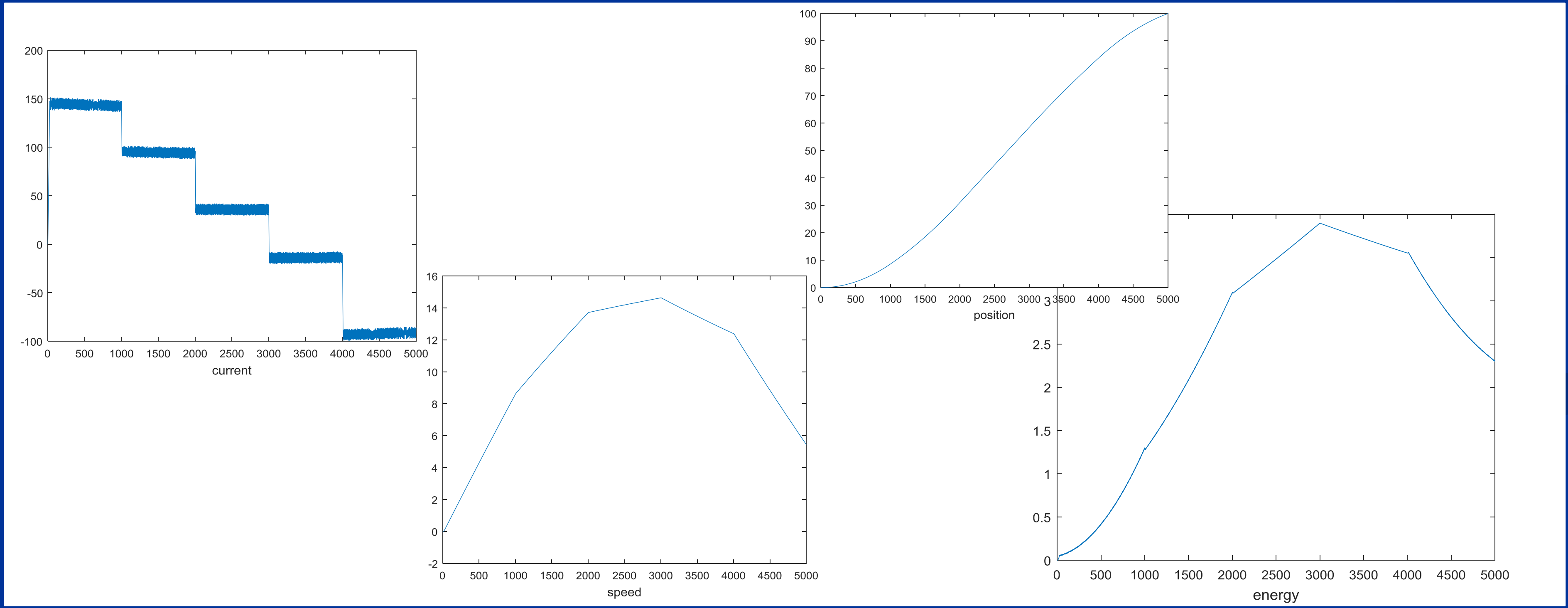

$$u(t) := \begin{cases} -1 & \text{if } i_m(t) > iref + \frac{\Delta}{2} \\ +1 & \text{if } i_m(t) < iref - \frac{\Delta}{2} \\ u(t^-) & \text{else.} \end{cases}$$

We built the corresponding simulator where *iref* is going in.

# Simulation is biutiful

Iref control:

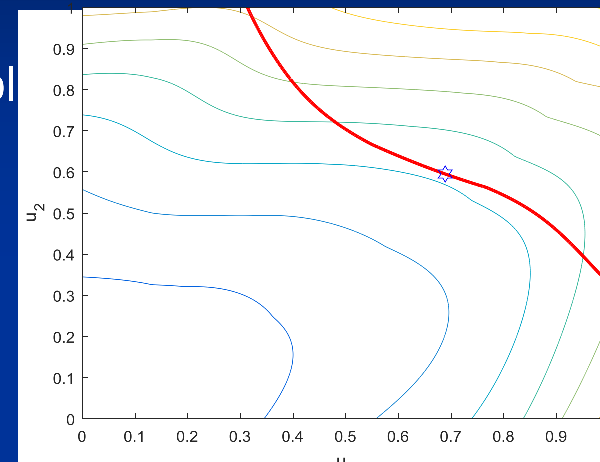




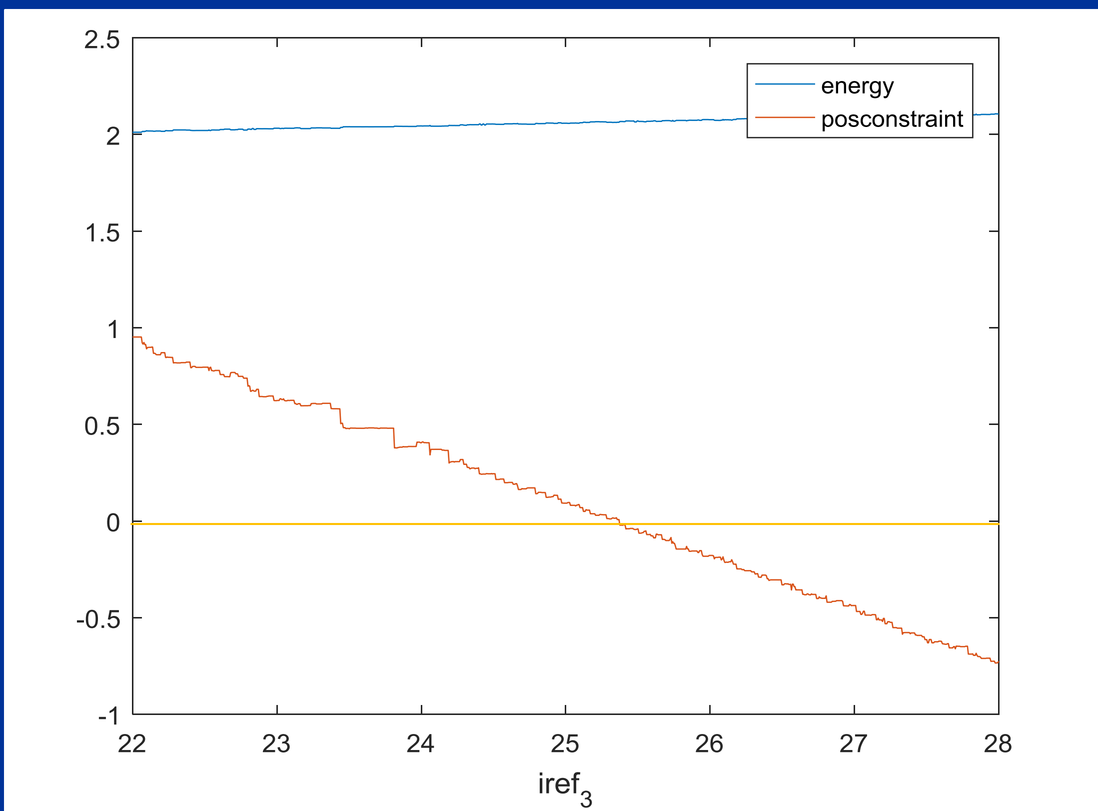
# Continuous optimisation is ugly
# Simulation based optimisation

Why does this give a numerical problem from nlp perspective in simulation based optimisation? No, pattern search neither works (well).

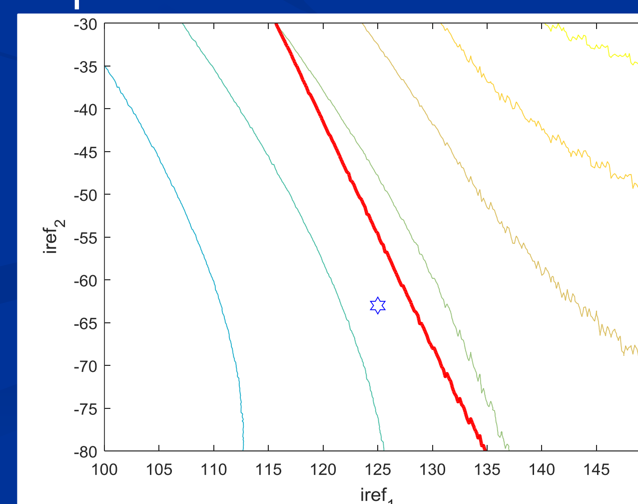
Compare to smooth control optimisation of $u_1$, $u_2$




Vary one variable $iref_3$; nonsmooth, non-continuous:

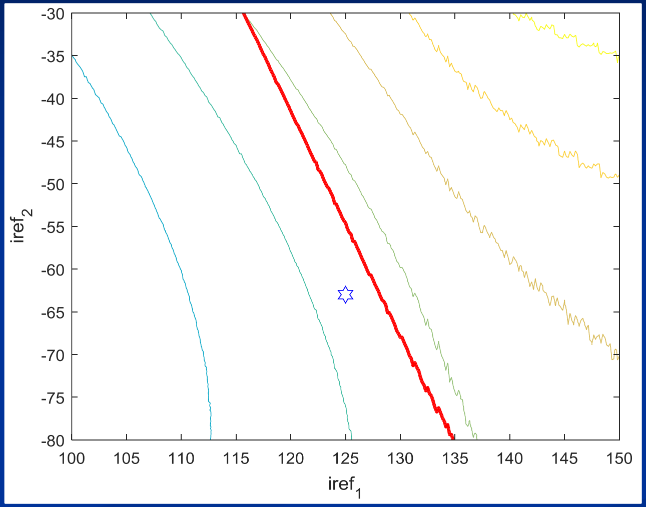



Optimum via GA



# Concluding

Kader did a discretisation of the iref space (grid) and used bounds
One could use a population algorithm.
Notice, it is not GO, just non-smooth




What to do:

- Let us focus on DP $u_t$ space
- Try DP in iref space
- Applying bounding, discretisation and interpolation