Advanced Control

March 7, 2014

École Centrale Paris, Châtenay-Malabry, France

Anne Auger

INRIA Saclay – Ile-de-France

Dimo Brockhoff

INRIA Lille – Nord Europe

Course Overview

Date		Topic
Fri, 10.1.2014	DB	Introduction to Control, Examples of Advanced Control
Fri, 17.1.2014	DB	Introduction to Fuzzy Logic
Fri, 24.1.2014	DB	Introduction to Artificial Neural Networks, Bio-inspired Optimization, discrete search spaces
Fri, 31.1.2014	AA	Continuous Optimization I
Fri, 7.2.2014	AA	Continuous Optimization II
break		
Fri, 28.2.2014	AA	The Traveling Salesperson Problem
Fr, 7.3.2014	DB	Controlling a Pole Cart
Fr, 14.3.2014		written exam (paper and computer)

all classes at 8h00-11h15 (incl. a 15min break around 9h30)

next week: exam at **8h00-11h15**

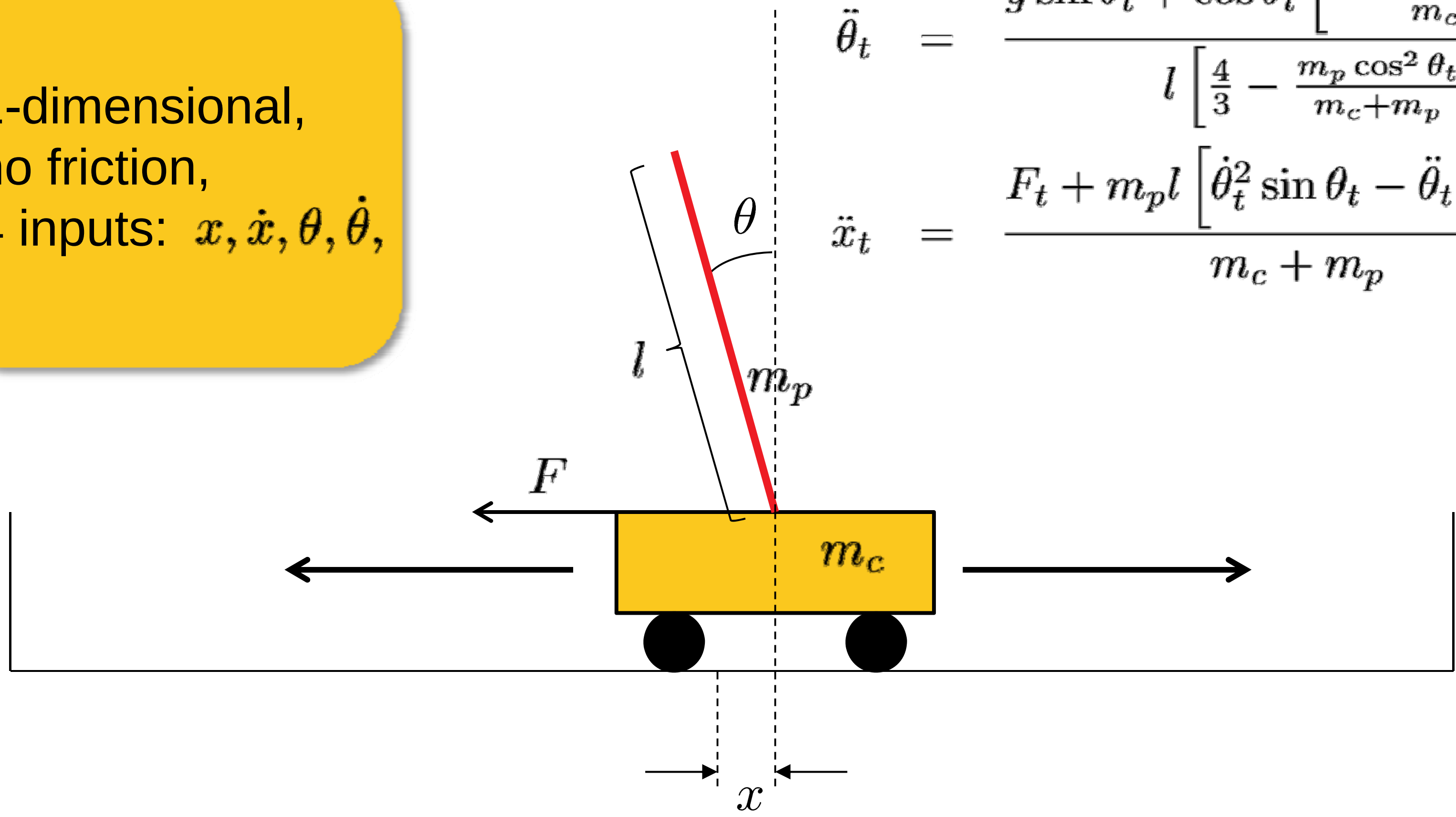
Exercise: Pole Balancing with ANNs and CMA-ES

Reminder: The Pole Balancing Benchmark

Typical benchmark example of a system with “advanced control”:

The Pole Balancing Problem

1-dimensional,
no friction,
4 inputs: $x, \dot{x}, \theta, \dot{\theta}$,



$$\ddot{\theta}_t = \frac{g\sin\theta_t + \cos\theta_t\left[\frac{-F_t - m_p l \dot{\theta}_t^2 \sin\theta_t}{m_c + m_p}\right]}{l\left[\frac{4}{3} - \frac{m_p \cos^2\theta_t}{m_c + m_p}\right]}$$

$$\ddot{x}_t = \frac{F_t + m_p l\left[\dot{\theta}_t^2 \sin\theta_t - \ddot{\theta}_t \cos\theta_t\right]}{m_c + m_p}$$

`http://researchers.lille.inria.fr/~brockhof/advancedcontrol/`

Reminder: Simulated Pole Balancing

Given all the parameters of the system, what do we do with it?

Answer: simulate!

- starting point: certain (random) position and angle; velocities and accelerations are zero
- choose discretization time step (e.g. $\tau = 0.02s$)
- at each time step, do:
  - compute $\ddot{\theta}_t$ with values $\dot{\theta}_t$ and θ_t
  - compute $\ddot{x}_t$ with $\dot{\theta}_t, \theta_t$ and the new $\ddot{\theta}_t$
  - $$\begin{aligned} x_{t+1} &= x_t + \tau \dot{x}_t \\ \dot{x}_{t+1} &= \dot{x}_t + \tau \ddot{x}_t \\ \theta_{t+1} &= \theta_t + \tau \dot{\theta}_t \\ \dot{\theta}_{t+1} &= \dot{\theta}_t + \tau \ddot{\theta}_t \end{aligned}$$

Reminder: Linear Control Law

Remark:

if the values and velocities of both position and angle are measured, there exists a linear (bang-bang) controller of the form:

$$F_t = F_m \, \mathrm{sgn}(k_1 x_t + k_2 \dot{x}_t + k_3 \theta_t + k_4 \dot{\theta}_t)$$

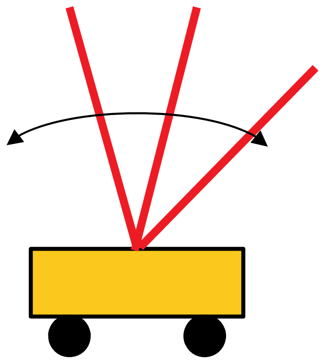
What we have seen:

random choice of k_1, k_2, k_3, k_4 enough to find a good controller most of the time

But

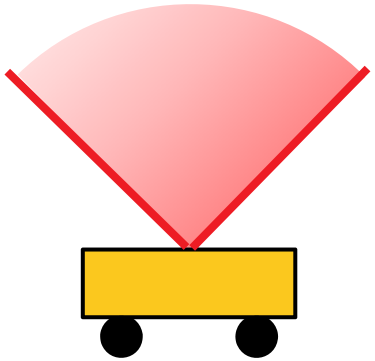
- this holds only for one specific initial condition of x_0 and θ_0
- parameters different for different initial conditions or random sampling of k_1, k_2, k_3, k_4 not enough anymore

Excursion: Robustness and Noise



A controller is robust if it works for different initial conditions - not only for one

→ simulate for different initial conditions



- however, amount of “testable” initial conditions is typically limited
- but one would like to find a controller that works for all initial conditions

→ simulate for different *random* conditions

random initialization introduces noisy measurements in terms of number of stable simulation steps

→ interested in *robust* solutions

More General Issue: Uncertainty

Uncertainty is always an important aspect in practice:

- the objective function is only a model of what we want

  measuring/simulation/modeling errors

- the problem formulation is static while reality is dynamic

  temperature, atmospheric pressure, ... changes

  material wears down

- even if we can detect the optimum, we might not be able to produce it

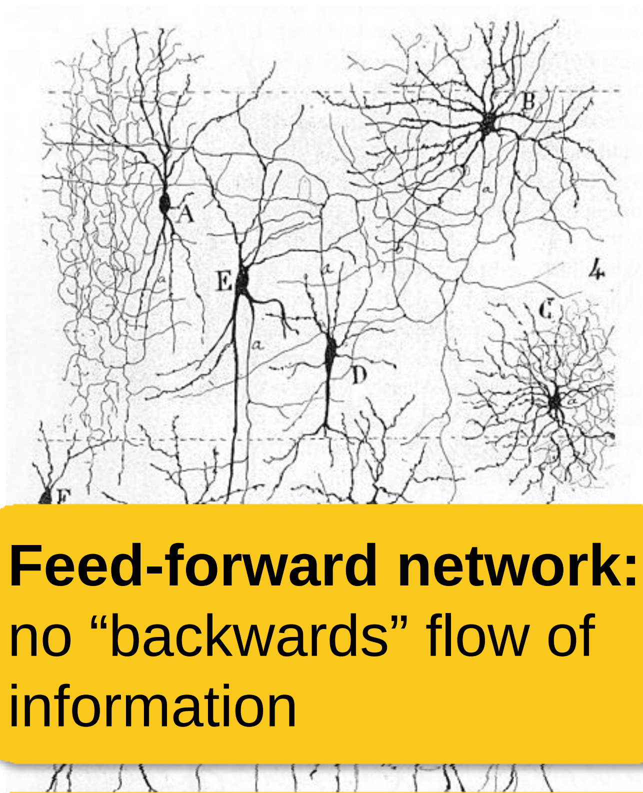
based on H.G Beyer and B. Sendhoff: “Robust Optimization – A Comprehensive Survey”. In Computer Methods in Applied Mechanics and Engineering, 196(33-34):3190-3218, 2007

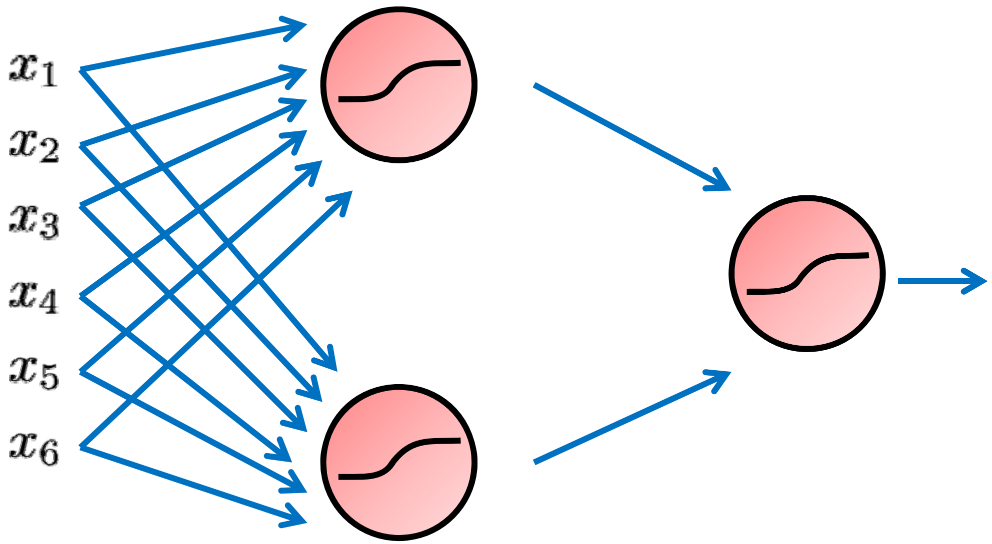
Exercise Part I:
Is the linear controller robust?

```
http://researchers.lille.inria.fr/~brockhof/advancedcontrol/
```

Combining Artificial Neurons

Artificial Neural Networks (ANNs) = a network of artificial neurons





Feed-forward network:
no “backwards” flow of information

Transfer functions:
output of each neuron based on inputs

$$y = \varphi\left(\sum_{i=1}^{n} w_i x_i\right)$$

Exercise Part II: Implementing an Artificial Neural Network

`http://researchers.lille.inria.fr/~brockhof/advancedcontrol/`

The Algorithm CMA-ES

Input: $\boldsymbol{m} \in \mathbb{R}^n$, $\sigma \in \mathbb{R}_+$, λ
Initialize: $\mathbf{C} = \mathbf{I}$, and $\boldsymbol{p}_\mathbf{c} = \mathbf{0}$, $\boldsymbol{p}_\sigma = \mathbf{0}$,
Set: $c_\mathbf{c} \approx 4/n$, $c_\sigma \approx 4/n$, $c_1 \approx 2/n^2$, $c_\mu \approx \mu_w/n^2$, $c_1 + c_\mu \le 1$, $d_\sigma \approx 1 + \sqrt{\frac{\mu_w}{n}}$,
and $w_{i=1\ldots\lambda}$ such that $\mu_w = \frac{1}{\sum_{i=1}^{\mu} w_i{}^2} \approx 0.3\,\lambda$
While not terminate

$$\boldsymbol{x}_i = \boldsymbol{m} + \sigma \boldsymbol{y}_i, \quad \boldsymbol{y}_i \sim \mathcal{N}_i(\mathbf{0}, \mathbf{C}), \quad \text{for } i = 1, \ldots, \lambda \qquad \text{sampling}$$

$$\boldsymbol{m} \leftarrow \sum_{i=1}^{\mu} w_i \boldsymbol{x}_{i:\lambda} = \boldsymbol{m} + \sigma \boldsymbol{y}_w \quad \text{where } \boldsymbol{y}_w = \sum_{i=1}^{\mu} w_i \boldsymbol{y}_{i:\lambda} \qquad \text{update mean}$$

$$\boldsymbol{p}_\mathbf{c} \leftarrow (1 - c_\mathbf{c})\boldsymbol{p}_\mathbf{c} + \mathbb{1}_{\{\|\boldsymbol{p}_\sigma\| < 1.5\sqrt{n}\}} \sqrt{1 - (1 - c_\mathbf{c})^2} \sqrt{\mu_w}\, \boldsymbol{y}_w \qquad \text{cumulation for } \mathbf{C}$$

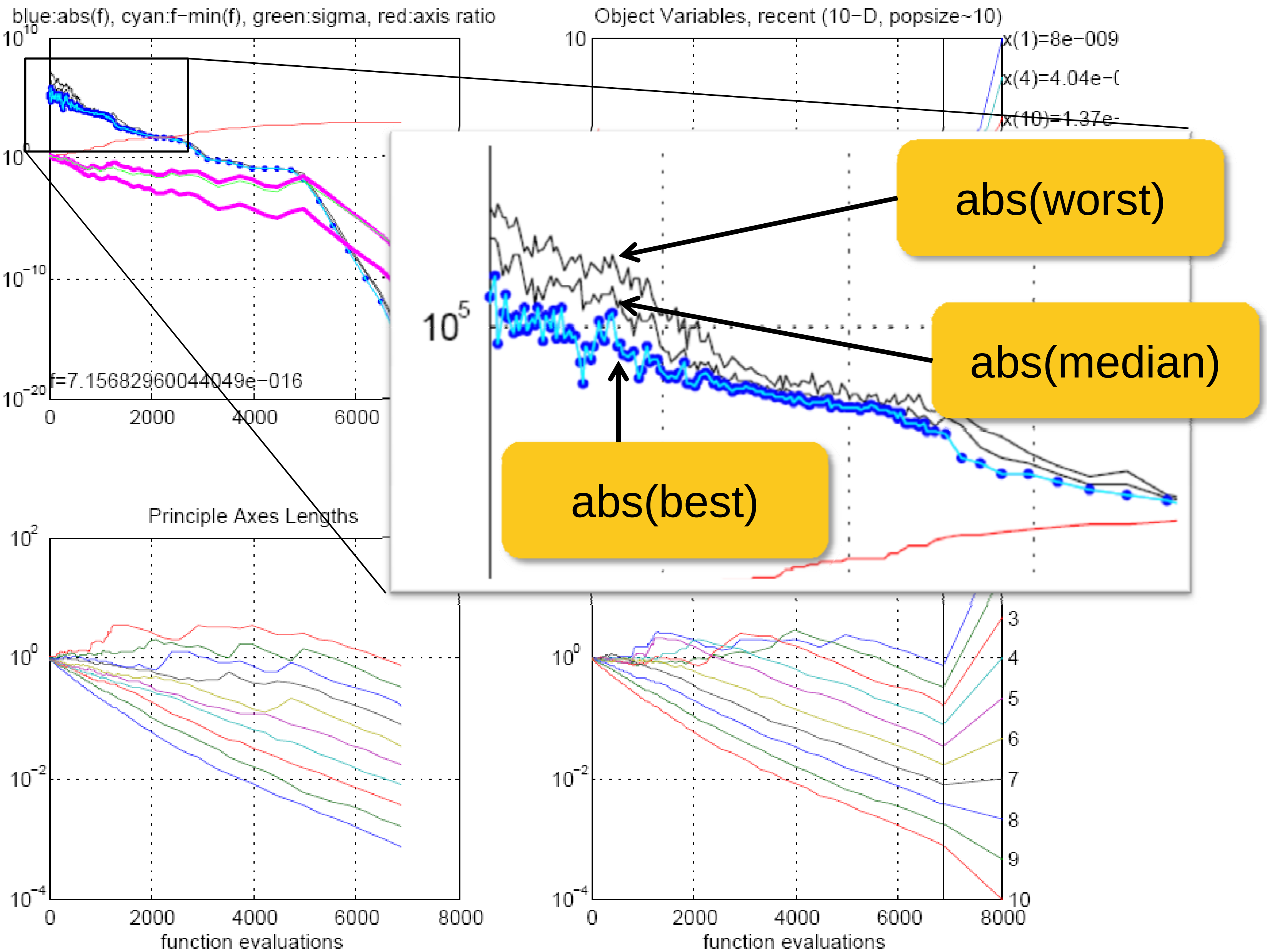
$$\boldsymbol{p}_\sigma \leftarrow (1 - c_\sigma)\boldsymbol{p}_\sigma + \sqrt{1 - (1 - c_\sigma)^2} \sqrt{\mu_w}\, \mathbf{C}^{-\frac{1}{2}} \boldsymbol{y}_w \qquad \text{cumulation for } \sigma$$

$$\mathbf{C} \leftarrow (1 - c_1 - c_\mu)\,\mathbf{C} + c_1 \boldsymbol{p}_\mathbf{c} \boldsymbol{p}_\mathbf{c}^\mathrm{T} + c_\mu \sum_{i=1}^{\mu} w_i \boldsymbol{y}_{i:\lambda} \boldsymbol{y}_{i:\lambda}^\mathrm{T} \qquad \text{update } \mathbf{C}$$

$$\sigma \leftarrow \sigma \times \exp\left(\frac{c_\sigma}{d_\sigma}\left(\frac{\|\boldsymbol{p}_\sigma\|}{\mathsf{E}\|\mathcal{N}(\mathbf{0},\mathbf{I})\|} - 1\right)\right) \qquad \text{update of } \sigma$$

Not covered on this slide: termination, restarts, useful output, boundaries and encoding

The output of CMA-ES



Issues on the Representation

Observation

- The weights of ANNs are typically normalized and lie within $[0, 1]$
- But CMA-ES does not restrict the variables in the standard setting

Hence, we have to set the bound constraints correctly:

```
opts.LBounds = 0;
opts.UBounds = 1;
```

Exercise Part III:
Using CMA-ES to Optimize the Weights of our ANN controller

http://researchers.lille.inria.fr/~brockhof/advancedcontrol/