Efficient Reinforcement Learning for Motor Control

Marc Peter Deisenroth

Computational and Biological Learning

UNIVERSITY OF CAMBRIDGE

joint work with Carl Edward Rasmussen

10th International PhD Workshop on Systems and Control

Hluboká nad Vltavou, Czech Republic

September 23, 2009

Why learning for control?

borrowed from www.harting-mitronics.ch

Figure: Robots assembling a car.

- machines can execute very complicated control commands

Why learning for control?

with permission from `http://www.chesshistory.com`

Figure: Kasparov (left) vs. DeepBlue (right), 1996/1997

- but sometimes control is not so easy

Why learning for control?

with permission from http://www.chesshistory.com

Figure: Kasparov (left) vs. DeepBlue (right), 1996/1997

- but sometimes control is not so easy

→ make machines solve control tasks themselves (learning)

Challenges in learning control

- machines typically require expert knowledge or many $(10^x,\, x \geq 2)$ trials
  ➡ can be a) expensive, b) not available, c) infeasible

Challenges in learning control

- machines typically require expert knowledge or many $(10^x,\ x \geq 2)$ trials
  ➡ can be a) expensive, b) not available, c) infeasible
- data-efficient
- make machines learn from "scratch"
  ➡ only general assumptions, no expert knowledge

Challenges in learning control

- machines typically require expert knowledge or many $(10^x,\, x \geq 2)$ trials
  ➡ can be a) expensive, b) not available, c) infeasible
- data-efficient
- make machines learn from "scratch"
  ➡ only general assumptions, no expert knowledge

objective:
➡ find a strategy of solving a problem that satisfies these constraints

Task learning as an optimal control problem

- find a policy/strategy π that yields low expected long-term cost

$$V^{\pi}(\mathbf{x}_0) = \sum_{t=0}^{T} \mathop{\mathbb{E}}_{\mathbf{x}_t}[c(\mathbf{x}_t)]$$

  of following policy π for T time steps (starting from $\mathbf{x}_0$)
- $c(\mathbf{x}_t)$: immediate/instantaneous cost function

Task learning as an optimal control problem

- find a policy/strategy π that yields low expected long-term cost

$$V^{\pi}(\mathbf{x}_0) = \sum_{t=0}^{T} \mathbb{E}_{\mathbf{x}_t}[c(\mathbf{x}_t)]$$

  of following policy π for T time steps (starting from $\mathbf{x}_0$)
- $c(\mathbf{x}_t)$: immediate/instantaneous cost function

challenges:

- data-efficient solution (few trials)
- unknown system function
- no expert knowledge available

Task learning as an optimal control problem

- find a policy/strategy π that yields low expected long-term cost

$$V^{\pi}(\mathbf{x}_0) = \sum_{t=0}^{T} \mathbb{E}_{\mathbf{x}_t}[c(\mathbf{x}_t)]$$

  of following policy π for T time steps (starting from $\mathbf{x}_0$)
- $c(\mathbf{x}_t)$: immediate/instantaneous cost function

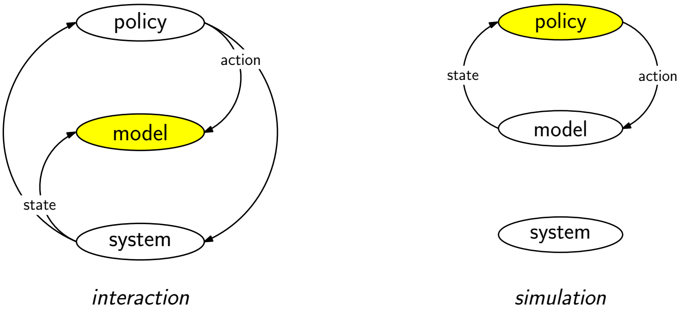
challenges:

- data-efficient solution (few trials)
- unknown system function
- no expert knowledge available

two possible approaches to get V^{π}:

- model free ➡ sample states and controls from real system
- model based ➡ find a model of the system function; internal simulation

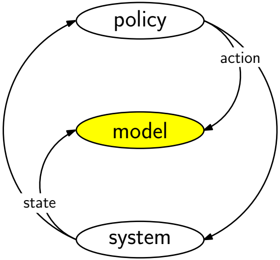
General (model-based) setup: interaction and simulation



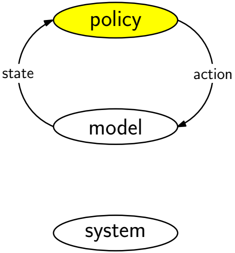
two phases

- interaction: internal model is refined using experience from interacting with the real system
- simulation: internal model is used to simulate consequences of actions in the real system, policy is refined

General (model-based) setup: interaction and simulation



interaction



simulation

two phases

- interaction: internal model is refined using experience from interacting with the real system
- simulation: internal model is used to simulate consequences of actions in the real system, policy is refined

➡ problem: model bias!

How do we get a good model?

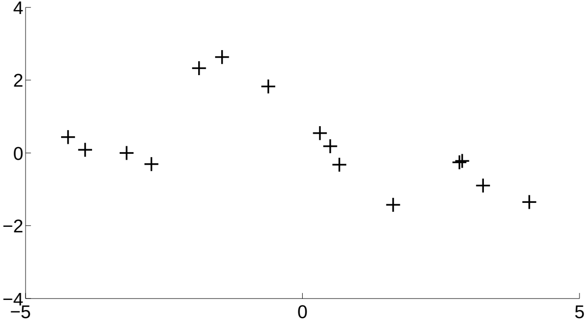
- system identification?

How do we get a good model?

- system identification?
- extract "shape" of the system function from data with high-level assumptions (e.g. smoothness)
- model what we know and what we don't

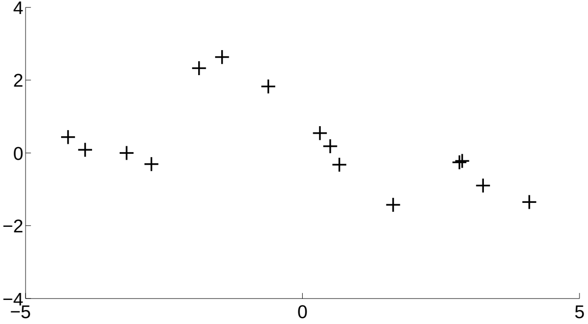
How do we get a good model?

- system identification?
- extract "shape" of the system function from data with high-level assumptions (e.g. smoothness)
- model what we know and what we don't



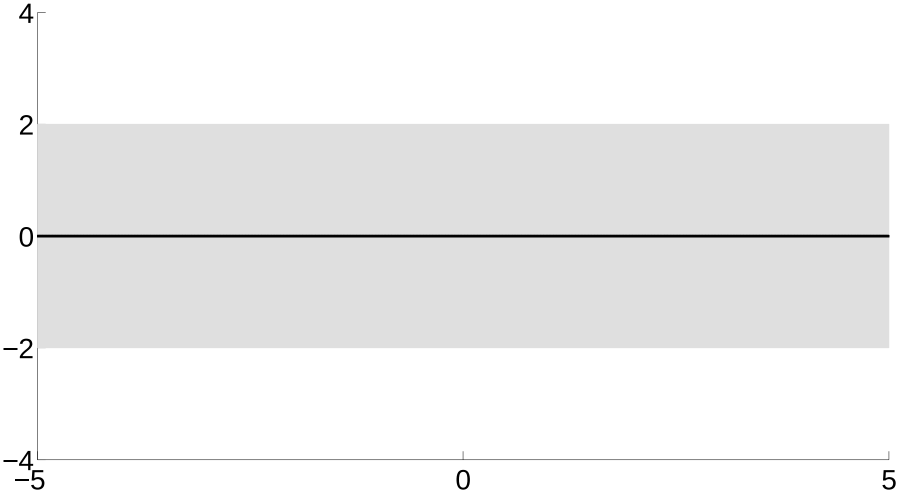
How do we get a good model?

- system identification?
- extract "shape" of the system function from data with high-level assumptions (e.g. smoothness)
- model what we know and what we don't

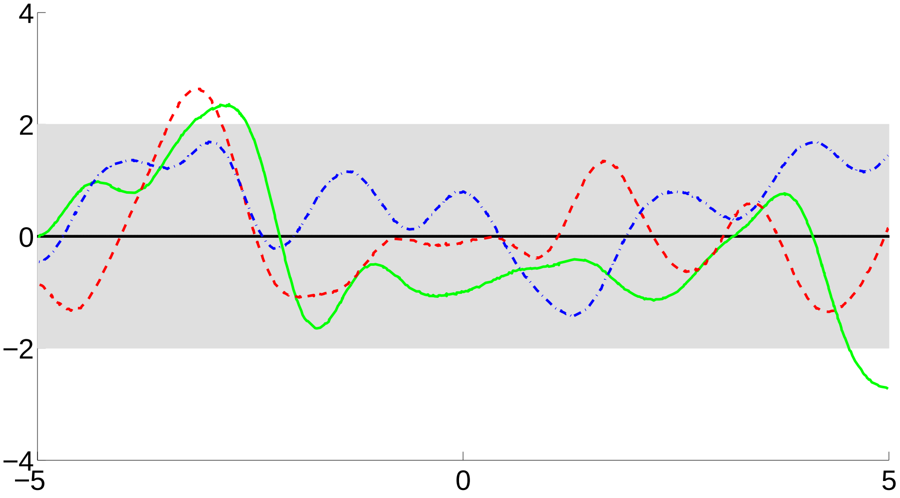


➡ here: Gaussian processes to find a model of the system function

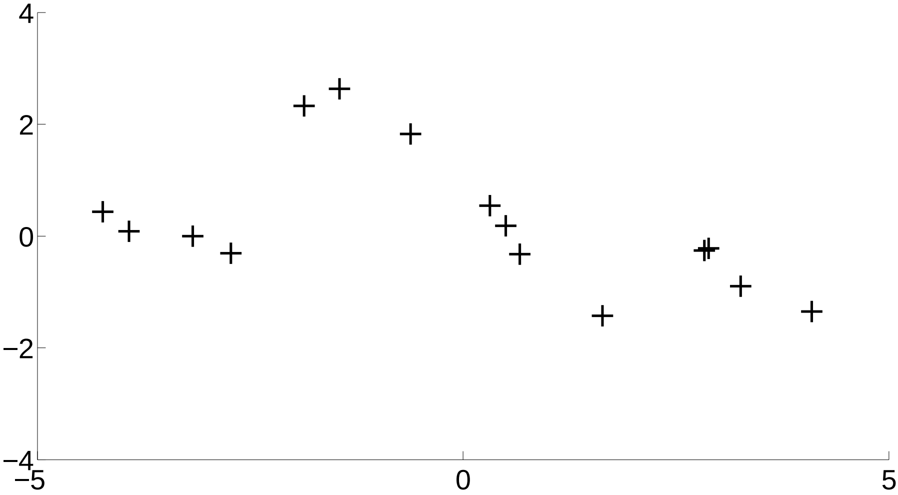
Pictorial introduction to Gaussian process regression



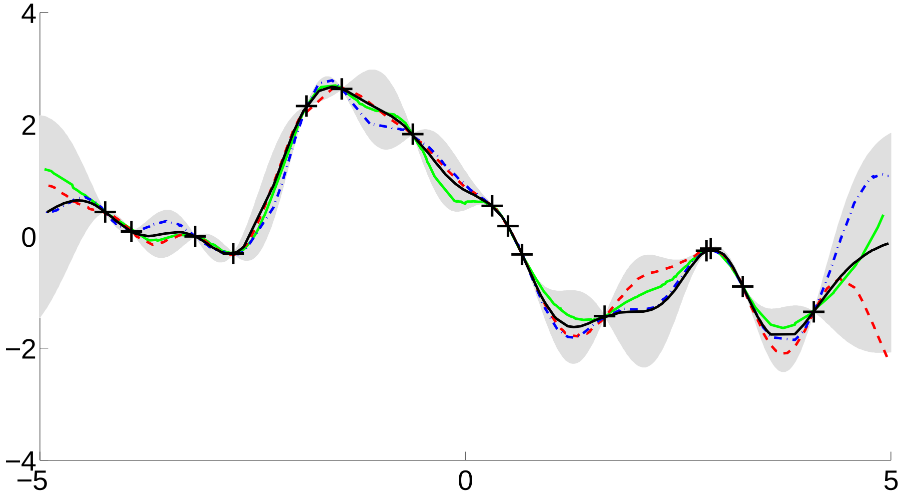
Pictorial introduction to Gaussian process regression



Pictorial introduction to Gaussian process regression



Pictorial introduction to Gaussian process regression



Evaluation of the value function

- the GP gives us one-step transition probabilities $p(\mathbf{x}_{t+1}|\mathbf{x}_t)$, but we need

$$V^\pi(\mathbf{x}_0) = \sum_{t=0}^{T} \mathbb{E}_{\mathbf{x}_t}[c(\mathbf{x}_t)]$$

Evaluation of the value function

- the GP gives us one-step transition probabilities $p(\mathbf{x}_{t+1}|\mathbf{x}_t)$, but we need

$$V^\pi(\mathbf{x}_0) = \sum_{t=0}^{T} \mathbb{E}_{\mathbf{x}_t}[c(\mathbf{x}_t)]$$

- cascade predictions to get $p(\mathbf{x}_1), p(\mathbf{x}_2), \dots, p(\mathbf{x}_T)$
- compute $\mathbb{E}_{\mathbf{x}_t}[c(\mathbf{x}_t)]$
- add them together

Policy refinement

- expected long-term cost (value function)

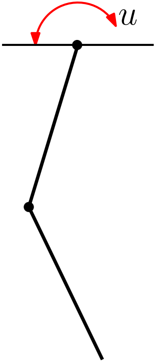
$$V^{\pi}(\mathbf{x}_0) = \sum_{t=0}^{T} \mathbb{E}_{\mathbf{x}_t}[c(\mathbf{x}_t)]$$

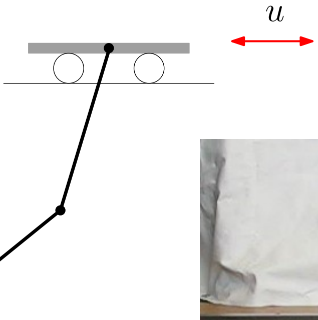
  can be evaluated analytically using approximate Bayesian inference
- compute derivative of $V^{\pi}(\mathbf{x}_0)$ with respect to policy parameters
- iterative gradient-based method to optimize policy parameters
  ➡ policy search

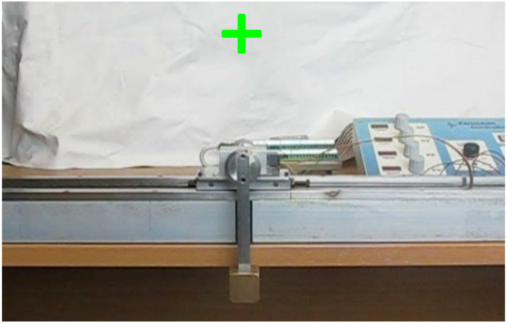
High-level algorithm

```
init: set policy to random
loop
    apply policy to the real system                    ▷ interaction
    learn GP model for system function
    loop                                               ▷ policy search
        simulate system with policy π                  ▷ predictions
        compute value function V^π for current policy
        improve policy                                 ▷ policy refinement
    end loop
end loop
```

Results







Wrap-up

- data-efficient artificial learning for control problems
- no expert knowledge
- probabilistic model for coherent representation of uncertainty
- explicit incorporation of uncertainty into prediction and decision-making
- gradient-based policy search
- works in simulation and hardware

http://mlg.eng.cam.ac.uk/marc