Learning Nonlinear State-Space Models for Control

Tapani Raiko and Matti Tornio

Helsinki University of Technology, Neural Networks Research Centre

Abstract

This paper studies the learning of nonlinear state-space models for a control task. This has some advantages over traditional methods. Variational Bayesian learning provides a framework where uncertainty is explicitly taken into account and system identification can be combined with model-predictive control. Three different control schemes are used. One of them, optimistic inference control, is a novel method based directly on the probabilistic modelling. Simulations with a cart-pole swing-up task confirm that the latent state space provides a representation that is easier to predict and control than the original observation space.

Nonlinear State-Space Models

- Modelling the dynamics of an unknown noisy system
- The dynamics $\mathbf{g}$ between hidden states $\mathbf{s}(t)$ and the observation mapping $\mathbf{f}$ from states are modelled as multi-layer perceptron (MLP) networks

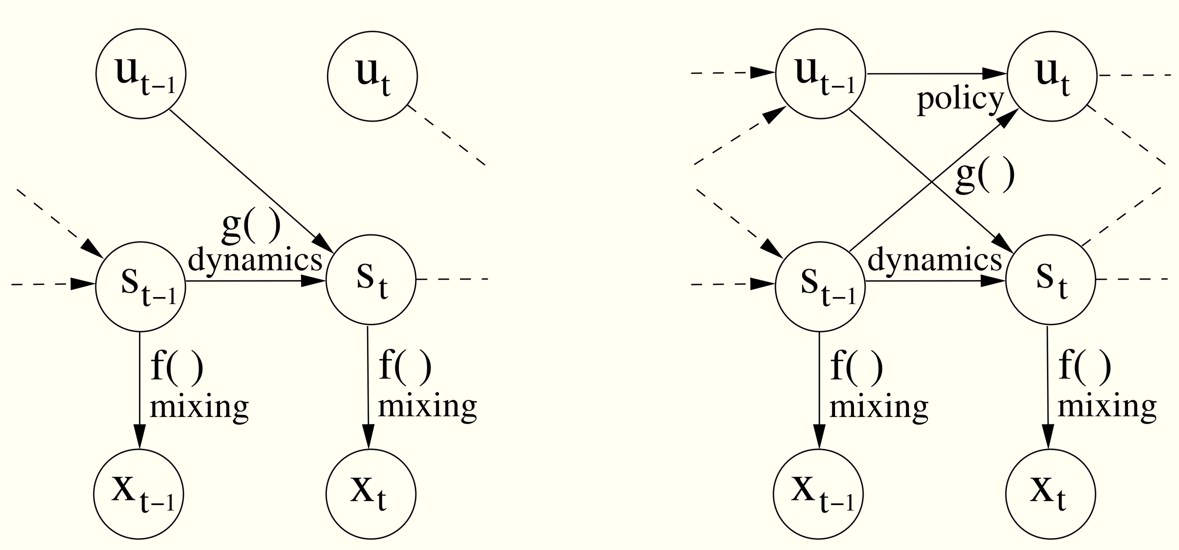
$$\mathbf{s}(t) = \mathbf{g}(\mathbf{s}(t-1), \boldsymbol{\theta}_\mathbf{g}) + \text{noise} \quad (1)$$

$$\mathbf{x}(t) = \mathbf{f}(\mathbf{s}(t), \boldsymbol{\theta}_\mathbf{f}) + \text{noise} \quad (2)$$

- The states $\mathbf{s}(t)$ and mappings $\mathbf{f}$ and $\mathbf{g}$ are learned from data
- Variational Bayesian learning avoids overfitting
  A parametric distribution over states and parameters is fitted to the true posterior
- H. Valpola and J. Karhunen. An unsupervised ensemble learning method for nonlinear dynamic state-space models. *Neural Computation*, 14(11):2647–2692, 2002.

Introducing Control Signals

- Two ways to introduce control signals (or actions) $\mathbf{u}(t)$ into the model



- We chose the one on the right for three reasons
  - allows three different control schemes
  - opportunity to find a task-oriented state space
  - biologically motivated (different parts of the cerebellum can be used for motor control and cognitive processing depending on where their outputs are directed)
- Equation (1) is replaced with

$$\begin{bmatrix} \mathbf{u}(t) \\ \mathbf{s}(t) \end{bmatrix} = \mathbf{g}\left(\begin{bmatrix} \mathbf{u}(t-1) \\ \mathbf{s}(t-1) \end{bmatrix}, \boldsymbol{\theta}_\mathbf{g} \right) + \text{noise} \quad (3)$$

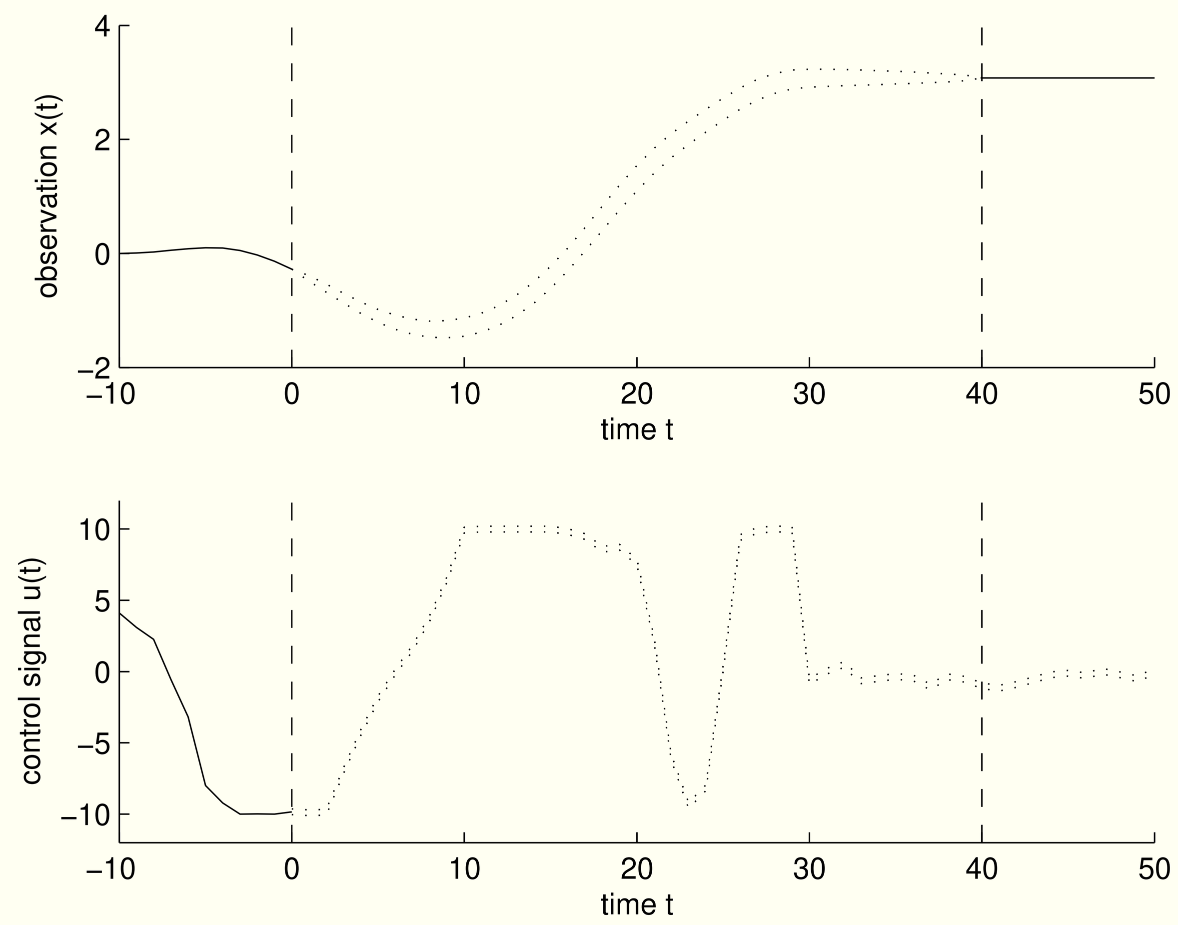
- Control schemes work at every time step:
  - Input: learned model, the history of observations and control signals
  - Output: control signal $\mathbf{u}(t_0)$ for the current time t_0

Direct Control (DC)

- The neural network acts as a controller
- Infer $\mathbf{s}(t)$ and use Eq. (3) to get $\mathbf{u}(t_0)$

Optimistic Inference Control (OIC)

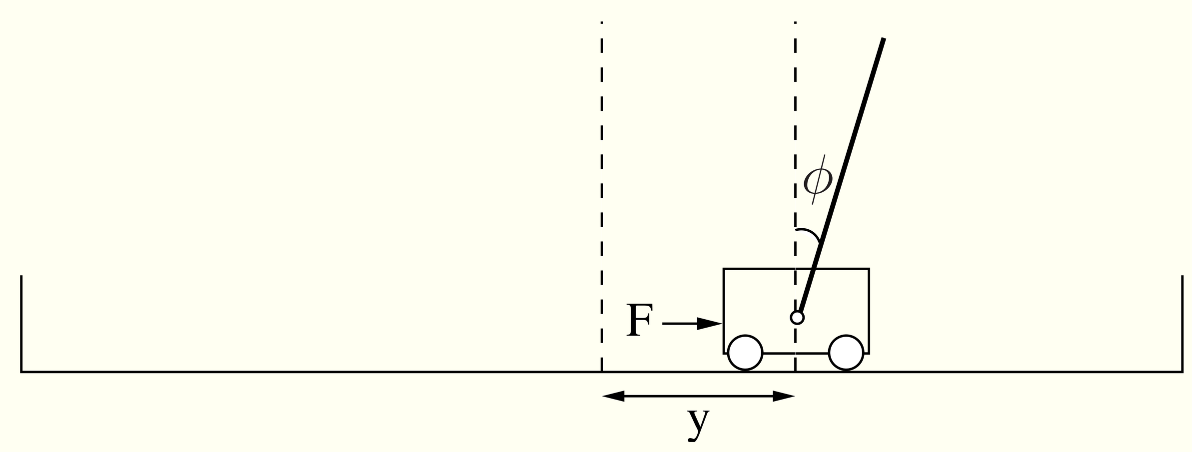
- Assume that the goal is reached after a fixed delay
- Infer what has happened in between (probabilistic smoothing)



Nonlinear Model Predictive Control (NMPC)

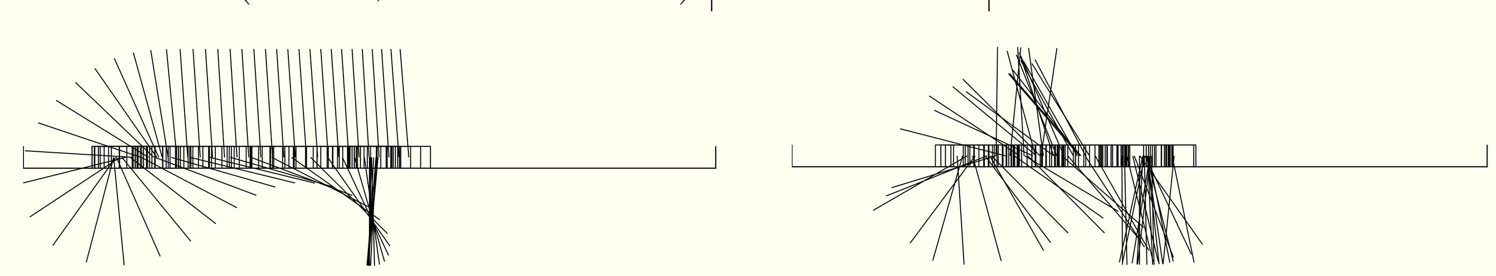
- Define a cost (or negated utility) function J over the future
- Find such a sequence of $\mathbf{u}$s that the expected cost $E\{J\}$ is minimised

Cart-Pole Swing-Up Task



- Observations: position y, the angle of the pole θ, and their velocities
- Control input: Force F
- System is unknown (learned from data)
- Goal is to swing the pole to an upward position and stabilise it without hitting the walls
- A comparison model $\mathbf{f} = \mathbf{I}$ does not have a hidden state:
  Equation (2) is replaced with $\mathbf{x}(t) = \mathbf{s}(t) + \text{noise}$
- Another comparison: Only y and θ observed (no velocities)
- Percentage of successful and partially successful swingups:

Setting	Little noise		Much noise	
Direct Control	14	48	4	31
Optimistic Inference Control	97	100	94	98
NMPC	100	100	94	95
NMPC (no velocities)	14	66	1	21
NMPC ($\mathbf{f} = \mathbf{I}$)	100	100	70	70
NMPC ($\mathbf{f} = \mathbf{I}$, no velocities)	0	0	0	0



Conclusion

- Nonlinear state-space models & three different control schemes
- Optimistic inference control reduces control into inference
- State-space models are resistant to noise
- Modelling the policy leads to task-oriented state representation
- Future work: faster algorithms, probing, exploration…
- Implementation and data: www.cis.hut.fi/projects/bayes/software