VARIATIONAL BAYESIAN APPROACH FOR NONLINEAR IDENTIFICATION AND CONTROL

Matti Tornio [*] **Tapani Raiko** [*]

[*] *Adaptive Informatics Research Centre, Helsinki University of Technology, P.O.Box 5400, FI-02015 TKK, Espoo, FINLAND*
{firstname.lastname}@hut.fi

Abstract: This paper studies the identification and model predictive control in nonlinear state-space models. Nonlinearities are modelled with neural networks and system identification is done with variational Bayesian learning. In addition to the robustness of control, the stochastic approach allows for a novel control scheme called optimistic inference control. We study the speed and accuracy of the two control schemes as well as the effect of changing horizon lengths and initialisation methods using a simulated cart-pole system. *Copyright © 2006 IFAC*

Keywords: nonlinear models, state-space models, stochastic control, neural network models, variational analysis, identification algorithms

1. INTRODUCTION

Learning is extremely important for control of complex systems (Åström et al., 2001). Nonlinear control is difficult even in the case that the system dynamics are known. If the dynamics are not known, the traditional approach is to make a model of the dynamics (system identification) and then try to control the simulated model (nonlinear model predictive control). The model learned from data is of course not perfect, but these imperfections are often ignored. The modern view of control sees feedback as a tool for uncertainty management (Murray et al., 2003), but managing it already in the modelling might have advantages. For instance, the controller can avoid regions where the confidence in model is not high enough (Kocijan et al., 2003).

The idea of learning probabilistic nonlinear state-space models for control is not new. The theory and different phenomena are already well covered in (Bar-Shalom, 1981). What has changed, though, is the range of models that can be used in practice, due to developments in Bayesian learning theory and computer performance. The issue remains challenging, i.e. investigation in (Bemporad et al., 2000) tells that observability and controllability properties of these systems cannot be easily deduced.

Recently, Rosenqvist and Karlström (2005) presented a method for system identification and control in nonlinear state-space models. The nonlinearities are modelled as piecewise linear (or affine). The system identification is based on the prediction error method. In probability theory, this corresponds to a maximum likelihood estimate assuming a Gaussian process noise. Our method continues this work by applying more sophisticated methods from the machine learning community.

Our method of choice, nonlinear dynamical factor analysis (NDFA) by Valpola and Karhunen (2002) is a state-of-the-art tool for finding nonlinear state-space models with variational Bayesian learning. In NDFA, the parameters, the states, and the observations are real-valued vectors that are modelled with parametrised probability distributions. Uncertainties from noisy observations and model imperfections are thus taken explicitly into account. Variational learning has many benefits compared to the maximum likelihood method. It is less prone to overfitting and can be used for selecting the model structure, e.g. the dimensionality of the state space.

An earlier version of our method is described in (Raiko and Tornio, 2005) from the neural-networks point of view. At the time we could not reasonable study the efficiency of the methods due to problems with the available inference algorithms. In this paper, we use a novel algorithm (Raiko et al., 2006) tailored for our purposes.

The rest of the paper is structured as follows: In Section 2, the nonlinear state-space estimator is reviewed and in Section 3 its use as a controller is presented. After experiments in Section 4 matters are discussed and concluded.

2. NONLINEAR STATE-SPACE MODELS

Selecting actions based on a state-space model instead of the observation directly has many benefits: Firstly, it is more resistant to noise (Raiko and Tornio, 2005) because it implicitly involves filtering. Secondly, the observations (without history) do not always carry enough information about the system state. Thirdly, when nonlinear dynamics are modelled by a function approximator such as an multilayer perceptron (MLP) network, a state-space model can find such a representation of the state that it is more suitable for the approximation and thus more predictable.

Nonlinear dynamical factor analysis (NDFA) (Valpola and Karhunen, 2002) is a powerful tool for system identification. It is based on a nonlinear state-space model learned in a variational Bayesian setting. NDFA scales only quadratically with the dimensionality of the observation space, so it is also suitable for modelling systems with fairly high dimensionality (Valpola and Karhunen, 2002).

In our model, the observation (or measurement) vector $\mathbf{y}(t)$ is assumed to have been generated from the hidden state vector $\mathbf{x}(t)$ driven by the control $\mathbf{u}(t)$ by the following generative model:

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

$$\mathbf{y}(t) = \mathbf{f}(\mathbf{x}(t), \boldsymbol{\theta}_\mathbf{f}) + \mathbf{w}(t) \quad (2)$$

where $\boldsymbol{\theta}$ is a vector containing the model parameters and time t is discrete. The process noise $\mathbf{v}(t)$ and the measurement noise $\mathbf{w}(t)$ are assumed to be independent, Gaussian, and white. Only the observations $\mathbf{y}$ are known beforehand, and both the states $\mathbf{x}$ and the mappings $\mathbf{f}$ and $\mathbf{g}$ are learned from the data. In the context of system identification this model can be considered task-oriented identification because of its internal forward model to predict $\mathbf{u}(t)$ (Raiko and Tornio, 2005). Note that the uncertainty of the process noise $\mathbf{v}(t)$ leaves the exact selection of the control signal $\mathbf{u}(t)$ open.

Multilayer perceptron (MLP) networks (Haykin, 1999) suit well to modelling both strong and mild nonlinearities. The MLP network models for $\mathbf{f}$ and $\mathbf{g}$ are

$$\mathbf{g}(\mathbf{x}(t), \boldsymbol{\theta}_\mathbf{g}) = \mathbf{x}(t) + \mathbf{B}\tanh\left[\mathbf{A}\mathbf{x}(t) + \mathbf{a}\right] + \mathbf{b} \quad (3)$$

$$\mathbf{f}(\mathbf{x}(t), \boldsymbol{\theta}_\mathbf{f}) = \mathbf{D}\tanh\left[\mathbf{C}\mathbf{x}(t) + \mathbf{c}\right] + \mathbf{d}, \quad (4)$$

where the sigmoidal tanh nonlinearity is applied component-wise to its argument vector. The parameters $\boldsymbol{\theta}$ include: (1) the weight matrices $\mathbf{A} \ldots \mathbf{D}$, the bias vectors $\mathbf{a} \ldots \mathbf{d}$; (2) the parameters of the distributions of the noise signals $\mathbf{w}(t)$ and $\mathbf{v}(t)$ and the column vectors of the weight matrices; (3) the hyperparameters describing the distributions of biases and the parameters in group (2).

There are infinitely many models that can explain any given data. In Bayesian learning, all the possible explanations are averaged weighting by their posterior probability. The posterior probability $p(\mathbf{x}, \boldsymbol{\theta} \mid \mathbf{y})$ of the states and the parameters after observing the data, contains all the relevant information about them. Variational Bayesian learning is a way to approximate the posterior density by a parametric distribution $q(\mathbf{x}, \boldsymbol{\theta})$. The misfit is measured by the Kullback-Leibler divergence:

$$C_\mathrm{KL} = \int q(\mathbf{x}, \boldsymbol{\theta}) \log \frac{q(\mathbf{x}, \boldsymbol{\theta})}{p(\mathbf{x}, \boldsymbol{\theta} \mid \mathbf{y})} d\boldsymbol{\theta} d\mathbf{x}, \quad (5)$$

that is, the closer q is to the true Bayesian posterior, the smaller the cost function.

The approximation q needs to be simple for mathematical tractability and computational efficiency. Variables are assumed to depend of each other in the following way:

$$q(\mathbf{x}, \boldsymbol{\theta}) = \prod_{t=1}^{T} \prod_{i=1}^{m} q(x_i(t) \mid x_i(t-1)) \prod_j q(\theta_j), \quad (6)$$

where m is the dimensionality of the state space $\mathbf{x}$. Furthermore, q is assumed to be Gaussian. To summarise, the distribution q is parametrised by the means and the variances of the unknown states and model parameters, and covariances of consecutive state components. The mean of a variable, say $\mathbf{x}(t)$, over the distribution q is marked with $E_q\left\{\mathbf{x}(t)\right\}$.

Inference (or state estimation) happens by adjusting the values corresponding to hidden states in q such that the cost function C_KL is minimised. Learning (or system identification) happens by adjusting both the hidden states and the model parameters in q minimising C_KL. The same cost function can also be used for determining the model structure, e.g. the dimensionality of the state space. The NDFA package contains an iterative minimisation algorithm for that. A good initialisation and other measures are essential to avoid getting stuck into a bad local minimum of the cost function. The standard initialisation for the learning is based on principal component analysis of the data augmented with embedding. Details can be found in (Valpola and Karhunen, 2002).

3. CONTROL SCHEMES

In this section, two different control schemes are presented. First we describe nonlinear predictive control (NMPC) in variational Bayesian setting, and then the optimistic inference control (OIC) scheme. Different strategies for generating control initialisations are also explored.

3.1 Nonlinear Model Predictive Control (NMPC)

Nonlinear model predictive control (NMPC) (Mayne et al., 2000) is based on minimising a cost function J defined over a future window of fixed length T_c. For example, the quadratic difference between the predicted future observations $\mathbf{y}$ and a reference signal $\mathbf{r}$ can be used:

$$J(\mathbf{y}(t_0), \mathbf{u}(t_0), \dots, \mathbf{u}(t_0 + T_c - 1)) = \sum_{\tau=1}^{T_c} |\mathbf{y}(t_0 + \tau) - \mathbf{r}|^2 . \tag{7}$$

Then J is minimised w.r.t. the control signals $\mathbf{u}$ and the first one $\mathbf{u}(t_0)$ is executed. Direct analogy to decision theory is revealed when the control cost J is interpreted as negative utility.

Here, the states and observations (but not control signals) are modelled probabilistically so we minimise the expected cost $E_q\{J\}$ (Bar-Shalom, 1981). The current guess $\mathbf{u}(t_0), \dots, \mathbf{u}(t_0 + T_c - 1)$ defines a probability distribution over future states and observations. This inference can be done with a single forward pass, when ignoring the internal forward model, that is, the dependency of the state on future control signals. In this case, it makes sense to ignore the forward model anyway, since the future control signals do not have to follow the learned policy.

Minimisation of $E_q\{J\}$ is done with a quasi-Newton algorithm (Nocedal and Wright, 1999). For that, the partial derivatives $Y^{t_2} = \partial \mathbf{y}(t_2)/\partial \mathbf{u}(t_1)$ for all $t_0 \leq t_1 < t_2 \leq t_0 + T_c$ must be computed. For a single input system we can simply apply the chain rule to arrive at the Jacobian

$$Y^{t_2} = F^{t_2} G^{t_2 - 1} \cdots G^{t_1 + 1} G^{t_1}, \tag{8}$$

where F^t and G^t are the Jacobians of the mappings f and g at the the time instant t. Dynamic programming can be used to efficiently compute these partial derivatives for multiple values of t_1 and t_2 in linear time. The extension of this result to multi-input systems is also relatively straightforward.

The use of a cost function makes NMPC very versatile. Costs for control signals and observations can be set for instance to restrict values within bounds etc. Expectations over a quadratic cost (Eq. 7) are easy to evaluate because: $E_q\left\{|\mathbf{y}(t) - \mathbf{r}|^2\right\} = |E_q\{\mathbf{y}(t)\} - \mathbf{r}|^2 + \sum_{i=1}^{n} \mathrm{Var}_q\{y_i(t)\}$, where n is the dimensionality of the observation space $\mathbf{y}$ and $\mathrm{Var}\{\cdot\}$ is the variance over the distribution q. The two terms are the nominal and stochastic part of the cost function. There is a direct analogy with dual control (Åström and Wittenmark, 1995) which means balancing between good control and small estimation errors. The usefulness of the decomposition is discussed in (Bar-Shalom, 1981).

3.2 Optimistic Inference Control

Optimistic inference control (OIC) as described by (Raiko and Tornio, 2005) and by (Attias, 2003) independently, works as follows. Assume that after a fixed delay T_c, the desired goal is reached. That is, (some components of) the observations $\mathbf{x}$ are at the desired level $\mathbf{r}$. Given this optimistic assumption and the observations and control signals so far, infer what happens in between. Then choose the expectation of $q(\mathbf{u}(t_0))$. It should be noted that whereas NMPC can be used with a wide variety of different models, OIC requires a probabilistic internal forward model.

OIC propagates the evidence in two directions, forwards from the current state and, additionally, the evidence backwards from the desired future. The inference is conceptually simple, but algorithmically difficult. The information from the future needs to flow through tens of nonlinear mappings $\mathbf{g}$ before it affects $\mathbf{u}(t_0)$. The OIC algorithm as presented in (Raiko and Tornio, 2005) only propagates information one step forward and backward in time for each iteration. To speed up this process, total derivatives described in (Raiko et al., 2006) are used to replace the partial derivatives, which leads to much faster propagation of information. Another alternative for fast inference is the Extended Kalman Smoother (Anderson and Moore, 1979), which unfortunately suffers from stability issues and it is therefore only used to initialise the OIC algorithm.

OIC in a nutshell:

Given observations $\dots, \mathbf{y}(t_0 - 2), \mathbf{y}(t_0 - 1)$ and control signals $\dots, \mathbf{u}(t_0 - 2), \mathbf{u}(t_0 - 1)$

```
1: Fix future $\mathbf{y}(t_0 + T_c) = \mathbf{y}(t_0 + T_c + 1) =$
      $= \cdots = \mathbf{r}$
2: Infer the distribution $q(\mathbf{u}(t), \mathbf{x}(t), \mathbf{y}(t))$ for all $t$
3: Select the mean of $q(\mathbf{u}(t_0))$ as the control signal
4: Observe $\mathbf{y}(t_0)$ and release $\mathbf{y}(t_0 + T_c)$
5: Increase $t_0$ and loop from 1
```

In case there are constraints for control signals or observations, they are forced after every inference iteration. If the horizon is set too short or the goal is otherwise overoptimistic, the method becomes unreliable. Even with a realistic goal, it is not in general guaranteed that the iteration will converge to the optimal control signal, as the iteration may get stuck in a local minimum. The inferred control signals can be validated by releasing the optimistic future and reinferring. If the future changes a lot, the control is unreliable.

3.3 Initialisation for Control

For nonlinear control tasks it is important that the initial estimate for control is good, otherwise the optimisation algorithm may get stuck in a local minimum or fail to converge in reasonable time. In many cases, the control signal from the previous time step can be used with quite good results. However, when a new control task starts or the goal is changed, or there are unexcepted changes in the system state, the previous control signal is often a poor choice.

A second option is to use random initialisations. If multiple different initialisations are used, this can be more robust than the first option. Unfortunately this is also much more time consuming because multiple control strategies must be computed for a single time step. If an internal forward model is available, a third option is also possible. The current system state can be propagated forward in time and the control signal from these predictions can be used as an initialisation.

4. EXPERIMENTS

Mechanical dynamical systems are easily understandable by people and thus illustrative as examples. We chose a simulated system to ease experimentation. To make the setting more difficult, the controllers do not have access to the simulation equations but have to identify an unknown system instead.

4.1 Cart-Pole Swing-Up Task

The Cart-Pole system (Kimura and Kobayashi) is a classic benchmark for nonlinear control. The system consist of a pole (which acts as an inverted pendulum) attached to a cart (Figure 1). The force applied to the cart can be controlled, and the goal is to swing the pole to an upward position and stabilise it. This must be accomplished without the cart crashing into the walls of the track.

The observed variables of the system are the position of the cart s, angle of the pole measured from the upward position ϕ, and their first derivatives s' and ϕ'. Control input is the force F applied to the cart. The detailed dynamics and constraints for the simulated cart-pole system can be found in (Kimura and Kobayashi).

A discrete system was simulated with a time step of $\Delta t = 0.05$s. The possible force was constrained between -10N and 10N, and the position between -3m and 3m. The system was initialised to a random state taken from the uniform distributions $s = [-1, 1]$, $s' = [-2, 2]$, $\phi = [\pi - 1, \pi + 1]$, $\phi' = [-3, 3]$.

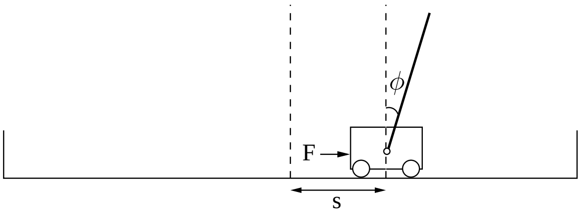

Fig. 1. The cart-pole system

4.2 Simulation

For all the simulations and the training data set additive Gaussian observation noise with $\sigma = 0.001$ and Gaussian process noise with $\sigma = 0.001$ were used. For the performance comparison between NMPC and OIC, the length of the control horizon was set to 40 time steps corresponding to 2 seconds of system's time. The simulations were run for 70 time steps corresponding to 3.5 seconds of system's time to ensure that the controller was able to stabilise the pole.

4.3 Implementation

The NDFA package (Valpola and Karhunen, 2002) version 1.0.0, the scripts for running the experiments, and the used training data are publicly available [1] .

During the training phase, data with 2500 samples was used. Most of the training data consisted of a sequence generated with semi-random control where the only goal was to ensure that the cart does not crash into the boundaries. Training data also contained some examples of hand-generated sections to better model the whole range of the observation and the dynamic mapping. The model was trained for 10000 iterations, which translates to several hours of computation time. Six-dimensional state space $\mathbf{x}(t)$ was used because it resulted in a model with the lowest cost function (Eq. 5).

The state $\mathbf{x}(t)$ was estimated using the iterated extended Kalman smoother (Anderson and Moore, 1979). A history of five observations and control signals seemed to suffice to give a reliable estimate. The reference signal $\mathbf{r}$ was $\phi = 0$ and $\phi' = 0$ at the end of the horizon and for five observations beyond that.

To take care of the constraints in the system with NMPC, a slightly modified version of the cost function (7) was used. Out-of-bounds values of the location of the cart and the force incurred a quadratic penalty, and the full cost function is

$$\begin{aligned} J_1(t_0, \mathbf{u}) = & J(t_0, \mathbf{u}) + \\ & \sum_{\tau=1}^{T_c} (\max(10, |u(t_0+\tau)|) - 10)^2 + \\ & \sum_{\tau=1}^{T_c} (\max(3, |y_s(t_0+\tau)|) - 3)^2, \end{aligned} \tag{9}$$

[1] http://www.cis.hut.fi/projects/bayes/software/

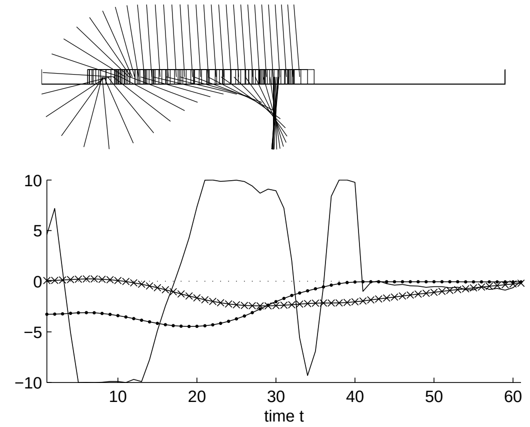

Fig. 2. Example of a successful swing-up with NMPC. Upper figure: visual representation of the swing-up. Lower figure: time series of the swing-up. Solid line is applied force, line with crosses is position of the cart and line with dots is the angle.

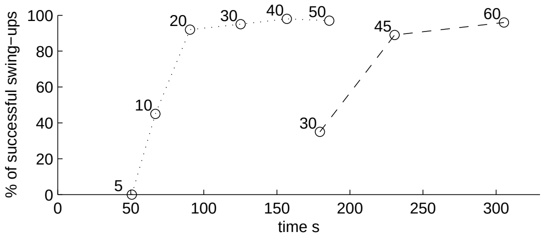

Fig. 3. Performance of the algorithms versus total computation time (in seconds). Dotted line is NMPC, dashed line is OIC. Numbers next to data points indicate number of iterations used. Control horizon length was 40 for all experiments.

where $y_s(t)$ refers to the location component s of the observation vector $\mathbf{y}(t)$.

4.4 Simulation Results

For all the control schemes, the cart-pole simulation was run for 100 times and the number of successful swing-ups was collected. As in (Kimura and Kobayashi), a swing-up is considered successful if the final angle is between -0.133π and 0.133π, final angular velocity between $-2\mathrm{rad/s}$ and $2\mathrm{rad/s}$, and the cart has not crashed into the boundaries of the area during swing-up.

Comparison of the performance of NMPC and OIC can be seen in Figure 3. With enough iterations, both methods reached a very high success rate. The few failed swing-ups were typically caused by difficult initial state of the cart-pole system resulting in an unfeasiable control strategy caused by limited horizon length. Example of a successful swing-up can be found in Figure 2.

On average, the traditional NMPC method was about 10 to 20 times slower than real-time on modern hardware (2.2 GHz AMD Opteron). The computation times for OIC were more varied, but in most cases the performance was inferior to NMPC. It should be noted, however, that the current implementation of OIC is quite heavily penalised by the presence of constraints, as the optimisation algorithm used cannot properly take their effects into account. In general, it is clear that further optimisations to the algorithms or improvements in hardware are required, before complex systems with fast dynamics can be controlled.

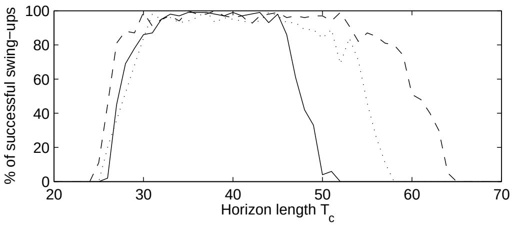

Fig. 4. The percentage of successful swing-ups as a function of the horizon length T_c in NMPC. Solid line is using old predictions as initialisation, dotted line is using initialisations based on the internal forward model and dashed line is using the best out of ten random initialisations. 50 NMPC iterations were used for all experiments.

The importance of the horizon length to the performance of the NMPC can be seen in Figure 4. All horizon lengths between 30 and 45 time steps had similar performance. Horizon lengths between 25 and 30 had problems with the cart crashing to the walls. Horizons shorter than 25 time steps could not reliably perform the swing-up task because the reference signal became too unrealistic.

Very long horizons are also problematic. First of all, they increase the computational burden of the algorithm. The increase in the number of the parameters often also leads into increase in the number of local minima, which makes the optimisation problem more involved. In addition, because only an approximative model of the system is available, predictions far to the future become more unreliable. This can lead the algorithm to choose an optimisation strategy which is not feasible in practice.

Different initialisations for the NMPC control signal show that local minima are the chief problem with long horizons (Figure 4). It was observed that in most failed swing-ups the controller made a large prediction error, and in the following time instant was unable to recover from the local minimum where both the force penalty and the reference signal tracking penalty both suddenly became large. A more reasonable way to generate new initialisations in such situations is to either use random initialisations or to use the internal forward model to generate a new control signal.

5. DISCUSSION AND CONCLUSION

Two different control schemes were studied in the framework of variational Bayesian learning of nonlinear state-space models. The first control scheme is stochastic nonlinear model predictive control, which is based on optimising control signals based on a cost function. The second scheme is optimistic inference control, which is based on fixing the desired observations at some point in the future and inferring the state and control signals between the future and the current state.

A controller might be able to carry out active information gathering or probing (Bar-Shalom, 1981). It means that in an unknown state, one should first decrease the uncertainty and then take action based on what has been revealed. Probing requires the controller to be able to plan to react to future observations. Optimistic inference control does this automatically in theory, but in practice it would require an even more sophisticated model for the posterior distribution than Equation 6. On a larger scale, to reduce the uncertainty of the model parameters, the controller should balance exploration and exploitation. A good starting point for taking exploration into account is in (Thrun, 1992).

Both control schemes presented here are computationally demanding. One possible way to speed up the NMPC algorithm would be to parallelise it. The MLP networks used in this work are not particularly well-suited for parallel computation, but many parts of the computation can still be divided to parts. The novel control scheme, OIC, provides a link between Bayesian inference and model-predictive control, but does not currently compete in efficiency.

Learning nonlinear state-space models seems promising for complex control tasks, where the observations about the system state are incomplete or the dynamics of the system is not well known. The experiments with a simple control task indicated the benefits of the proposed approach. There is still work left in combating high computational complexity and in giving some guarantees on performance especially in unexpected situations or near boundaries.

6. ACKNOWLEDGEMENTS

The authors would like to thank Janne Paanajärvi, Heikki Hyötyniemi, Kai Zenger, Sampsa Laine, Harri Valpola, and Markus Harva for fruitful discussions. This work was supported in part by the Finnish Centre of Excellence Programme (2000-2005) under the project New Information Processing Principles and by the IST Programme of the European Community under the PASCAL Network of Excellence, IST-2002-506778. This publication only reflects the authors' views.

REFERENCES

B. Anderson and J. Moore. *Optimal Filtering*. Prentice-Hall, Englewood Cliffs, NJ, 1979.

K.J. Åström and B. Wittenmark. *Adaptive Control*. Addison-Wesley Publishing, second edition, 1995. ISBN 0201558661.

K.J. Åström, P. Albertos, M. Blamke, A. Isidori, W. Schaufelberger, and R. Sanz. *Control of complex systems*. Springer, 2001. ISBN 1-85233-324-3.

H. Attias. Planning by probabilistic inference. In *Proc. AI-Stats 2003*, 2003.

Y. Bar-Shalom. Stochastic dynamic programming: Caution and probing. *IEEE Transactions on Automatic Control*, 26(5):1184–1195, October 1981.

A. Bemporad, G. Ferrari-Trecate, and M. Morari. Observability and controllability of piecewise affine and hybrid systems. *IEEE transactions on automatic control*, 45(10):1864–1876, 2000.

S. Haykin. *Neural Networks – A Comprehensive Foundation, 2nd ed.* Prentice-Hall, 1999.

H. Kimura and S. Kobayashi. Efficient non-linear control by combining Q-learning with local linear controllers. In *Proc. ICML*.

J. Kocijan, R. Murray-Smith, C. E. Rasmussen, and B. Likar. Predictive control with Gaussian process models. In *Proceedings of IEEE Region 8 Eurocon 2003: Computer as a Tool*, pages 352–356, 2003.

D.Q. Mayne, J.B. Rawlings, C.V. Rao, and P.O.M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36:789–814, 2000.

R. Murray, K. J. Åström, S. P. Boyd, R. W. Brockett, and G. Stein. Future directions in control in an information-rich world. *IEEE Control Systems Magazine*, 23(2):20–33, April 2003.

J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer-Verlag, New York, 1999.

T. Raiko and M. Tornio. Learning nonlinear state-space models for control. In *Proc. IJCNN'05*, pages 815–820, Montreal, Canada, 2005.

T. Raiko, M. Tornio, A. Honkela, and J. Karhunen. State inference in variational Bayesian nonlinear state-space models. In *Proc. ICA 2006*, pages 222–229, Charleston, South Carolina, USA, 2006.

F. Rosenqvist and A. Karlström. Realisation and estimation of piecewise-linear output-error models. *Automatica*, 41(3):545–551, March 2005.

S. B. Thrun. The role of exploration in learning control. In *Handbook of Intelligent Control: Neural, Fuzzy and Adaptive Approaches*, pages 527–559. Van Nostrand Reinhold, 1992.

H. Valpola and J. Karhunen. An unsupervised ensemble learning method for nonlinear dynamic state-space models. *Neural Computation*, 14(11):2647–2692, 2002.