Acta Cybernetica **18** (2008) 637–649.

InfoMax Bayesian Learning of the Furuta Pendulum

László A. Jeni[*], György Flórea[†], and András Lőrincz[†‡]

Abstract

We have studied the InfoMax (D-optimality) learning for the two-link Furuta pendulum. We compared InfoMax and random learning methods. The InfoMax learning method won by a large margin, it visited a larger domain and provided better approximation during the same time interval. The advantages and the limitations of the InfoMax solution are treated.

Keywords: Online Bayesian learning, D-optimality, infomax control, Furuta pendulum

1 Introduction

In recent years, machine learning methods became more and more accurate and popular, so that the task of learning the dynamics of plants and the learning of reactive behaviours to environmental changes seem to be within reach. Such tasks call for online (i.e., real time) learning methods. For fast learning, one would like to provide stimuli that facilitate the fastest information gain about the changes of the plant and its environment [3, 2].

As an example, consider an industrial robot. Programming of industrial robots is traditionally an off-line task. In typical situations, the trajectory of the robot is generated from a CAD model of the environment and the robot [12]. However, this model holds no information about the unavoidable modelling errors especially if the environment changes. We assume that the environment and the robot have a parametrised representation and that the goal is to estimate these parameters as quickly as possible and possibly on-the-flight.

An attractive route replaces trajectory planning and trajectory tracking with speed-field planning and speed-field tracking. This latter is less strict about the actual path. The difference between the two methods can be described by the example when one is walking on a crowded street. Here, the trajectory should be

[*] Eötvös Loránd University, Department of Software Technology and Methodology, E-mail: `jedi@inf.elte.hu`

[†] Eötvös Loránd University, Department of Information Systems, E-mail: `ripps11@freemail.hu`, `andras.lorincz@elte.hu`

[‡] Corresponding author

replanned at each time instant and at full length when anybody moves. Speed-field, however, undergoes slight changes even if the walker is pushed by the crowd. Further, speed-field tracking requires a crude model of the inverse dynamics and still has attractive global stability properties [13, 14].

This underlines our approach, where we try to approximate the dynamics fast. We use the InfoMax (also called D-optimality) approach. InfoMax learning means that the next control action is chosen according to Bayesian estimation given what we have learnt until *now*, given the actual state that we have *at this moment*. The task is the estimation of the best control signal *now* to gain the most information about the unknown parameters of the model from the next observation. Note, however, that the state of the unknown plant is also an issue, we may not know the order of the dynamics, or the temporal convolution corrupting instantaneous control actions. We treat the problem of the order of the dynamics here.

It has been shown recently by Póczos and Lőrincz [11] that InfoMax control can be computed analytically without approximations and leads to simple learning rules by slightly modifying the generalised linear model in [8].

Our particular example that we study is the so called *two-link Furuta pendulum* [4]. This pendulum as well as the related InfoMax task are sketched in Fig. 1.

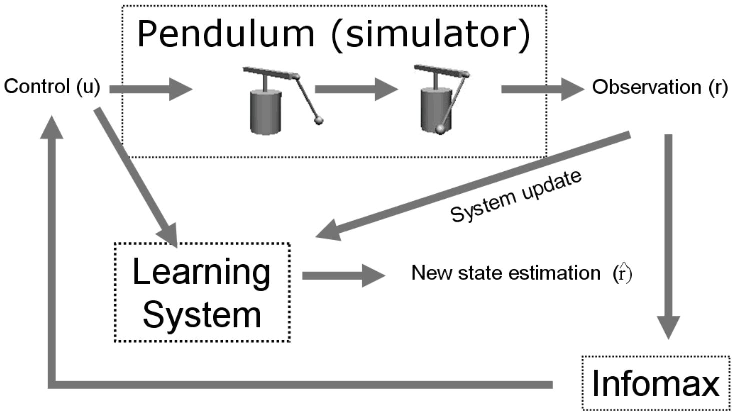


Figure 1: Furuta pendulum and InfoMax learning.
The pendulum stimulator receives the control input from the InfoMax algorithm, the only output of this exploratory algorithm. The learning system also receives this control signal. The learning system estimates, the pendulum stimulator computes the next state. All information, i.e., state, state estimation, control signal are used to update the parameters of the learning system and to compute the next control signal by the InfoMax algorithm.

The paper is organised as follows. In Section 2, we review the InfoMax approach. Section 3 is about the dynamics and the parametrisation of the Furuta pendulum problem. Section 4 describes the results. We summarise our findings and draw conclusions in Section 5.

2 Infomax Learning

We introduce the model used in [11]. Let us assume that we have d simple computational units called '*neurons*' in a recurrent neural network:

$$r_{t+1} = g\left(\sum_{i=0}^{I} F_i r_{t-i} + \sum_{j=0}^{J} B_j u_{t+1-j} + e_{t+1}\right), \tag{1}$$

where $\{e_t\}$, the driving noise of the RNN, denotes temporally independent and identically distributed (i.i.d.) stochastic variables and $P(e_t) = \mathcal{N}_{e_t}(0, V)$, $r_t \in \mathcal{R}^d$ represents the observed activities of the neurons at time t. Let $u_t \in \mathbb{R}^c$ denote the control signal at time t. The neural network is formed by the weighted delays represented by matrices F_i $(i = 0, \ldots, I)$ and B_j $(j = 0, \ldots, J)$, which connect neurons to each other and also the control components to the neurons, respectively. Control can also be seen as the means of interrogation, or the stimulus to the network [8]. We assume that function $g : \mathbb{R}^d \to \mathbb{R}^d$ in (1) is known and invertible. The computational units, the neurons, sum up weighted previous neural activities as well as weighted control inputs. These sums are then passed through identical non-linearities according to (1). The goal is to estimate the parameters $F_i \in \mathbb{R}^{d\times d}$ $(i = 0, \ldots, I)$, $B_j \in \mathbb{R}^{d\times c}$ $(j = 0, \ldots, J)$ and the covariance matrix V, as well as the driving noise e_t by means of the control signals.

We introduce the following notations:

$$x_{t+1} = [r_{t-I}; \ldots; r_t; u_{t-J+1}; \ldots; u_{t+1}], \tag{2}$$
$$y_{t+1} = g^{-1}(r_{t+1}), \tag{3}$$
$$A = [F_I, \ldots, F_0, B_J, \ldots, B_0] \in \mathbb{R}^{d\times m}. \tag{4}$$

Using these notations, the original model (1) reduces to a linear equation:

$$y_t = Ax_t + e_t. \tag{5}$$

The InfoMax learning relies on Bayes' method in the online estimation of the unknown quantities (parameter matrix A, noise e_t and its covariance matrix V). It assumes that prior knowledge is available and it updates the posteriori knowledge on the basis of the observations. Control will be chosen at each instant to provide maximal expected information concerning the quantities we have to estimate. Starting from an arbitrary prior distribution of the parameters the posterior distribution needs to be computed. This estimation can be highly complex, so approximations are common in the literature. For example, assumed density filtering, when the computed posterior is projected to simpler distributions, has been suggested [1, 10, 9]. Póczos and Lőrincz [11] used the method of conjugated priors [6], instead. For matrix A we assume a matrix valued normal (i.e., Gaussian) distribution prior. For covariance matrix V inverted Wishart (IW) [7] distribution will be our prior. There are three advantages of this choice: (i) they are somewhat more general than the typical Gaussian assumption, (ii) the functional form of the posteriori distributions is not affected, and (iii) the model (2-4) admits analytical – i.e., approximation-free – solution for this case as shown in [11]. Below, we review the main concepts and the crucial steps of this analytical solution.

Let us define the normally distributed matrix valued stochastic variable $A \in \mathbb{R}^{d\times m}$ by using the following quantities: $M \in \mathbb{R}^{d\times m}$ is the expected value of A. $V \in \mathbb{R}^{d\times d}$ is the covariance matrix of the rows, and $K \in \mathbb{R}^{m\times m}$ is the so-called precision parameter matrix that we shall modify in accordance with the Bayesian update. They are both positive semi-definite matrices.

Now, one can rewrite model (5) as follows:

$$\begin{aligned} P(A|V) &= \mathcal{N}_A(M,V,K), & (6)\\ P(V) &= \mathcal{IW}_V(Q,n), & (7)\\ P(e_t|V) &= \mathcal{N}_{e_t}(0,V), & (8)\\ P(y_t|A,x_t,V) &= \mathcal{N}_{y_t}(Ax_t,V). & (9)\end{aligned}$$

We introduce the following quantities:

$$\begin{aligned} \gamma_{t+1} &= 1 - x_{t+1}^T(x_{t+1}x_{t+1}^T + K_t)^{-1}x_{t+1},\\ n_{t+1} &= n_t + 1,\\ M_{t+1} &= (M_tK_t + y_{t+1}x_{t+1}^T)(x_{t+1}x_{t+1}^T + K_t)^{-1},\\ Q_{t+1} &= Q_t + (y_{t+1} - M_tx_{t+1})\,\gamma_{t+1}\,(y_{t+1} - M_tx_{t+1})^T\,, & (10)\end{aligned}$$

for the posterior probabilities. Then – one can show [11] that –

$$\begin{aligned} P(A|V,\{x\}_1^{t+1},\{y\}_1^{t+1}) &= \mathcal{N}_A(M_{t+1},V,x_{t+1}x_{t+1}^T + K_t), & (11)\\ P(V|\{x\}_1^{t+1},\{y\}_1^{t+1}) &= \mathcal{IW}_V\left(Q_{t+1},n_{t+1}\right), & (12)\\ P(y_{t+1}|\{x\}_1^{t+1},\{y\}_1^{t}) &= \mathcal{T}_{y_{t+1}}\left(Q_t,n_t,M_tx_{t+1},\gamma_{t+1}\right). \end{aligned}$$

The derivations give rise to a strikingly simple optimal control value expression:

$$u_{t+1^{opt}} = \arg\max_{u\in\mathcal{U}} x_{t+1}^T K_t^{-1} x_{t+1}, \tag{13}$$

The steps of the InfoMax update are summarised in Algorithm 1. We shall follow these steps in our computer studies on the Furuta pendulum that we detail in the next section.

3 The Furuta pendulum

3.1 Furuta Pendulum

The Furuta pendulum is shown in Figure 2. The pendulum has two links [4, 5]. Configuration of the pendulum is determined by the length of the links and by two

Algorithm 1 Pseudocode of the InfoMax algorithm.

Control Calculation

```
$u_{t+1} = \arg\max_{u \in \mathcal{U}} \hat{x}_{t+1}^T K_t^{-1} \hat{x}_{t+1}$
where $\hat{x}_{t+1} = [r_{t-I}; \ldots; r_t; u_{t-J+1}; \ldots; u_t; u]$
set $x_{t+1} = [r_{t-I}; \ldots; r_t; u_{t-J+1}; \ldots; u_t; u_{t+1}]$
```

Observation

```
observe $r_{t+1}$, and let $y_{t+1} = g^{-1}(r_{t+1})$
```

Bayesian update

```
$M_{t+1} = (M_t K_t + y_{t+1} x_{t+1}^T)(x_{t+1} x_{t+1}^T + K_t)^{-1}$
$K_{t+1} = x_{t+1} x_{t+1}^T + K_t$
$n_{t+1} = n_t + 1$
$\gamma_{t+1} = 1 - x_{t+1}^T (x_{t+1} x_{t+1}^T + K_t)^{-1} x_{t+1}$
$Q_{t+1} = Q_t + (y_{t+1} - M_t x_{t+1}) \gamma_{t+1} (y_{t+1} - M_t x_{t+1})^T$
```

angles. Dynamics of the pendulum are also determined by the different masses, i.e. the masses of the links and the mass of the end effector as well as by the two actuators, which are able to rotate the horizontal link and the swinging link in both directions, respectively. The angle of the horizontal link is denoted by ϕ, whereas the symbol for the angle of the horizontal link is θ (Fig. 2). Parameters of our computer studies are provided in Table 1. The state of the pendulum is given by ϕ, θ, $\dot{\phi}$ and $\dot{\theta}$. The magnitude of the angular speeds $\dot{\phi}$ and $\dot{\theta}$ was restricted to 2 rotations/s, i.e. to the interval $[-2\frac{\text{rot}}{\text{s}}, 2\frac{\text{rot}}{\text{s}}]$.

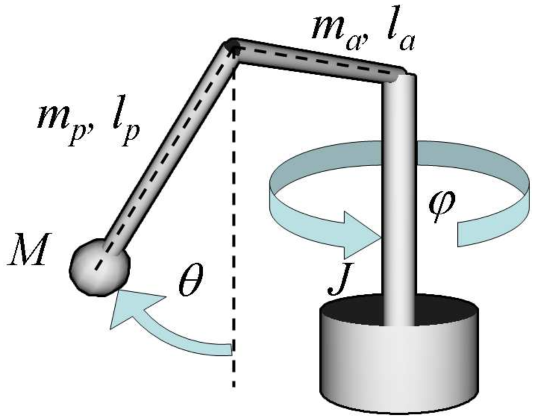


Figure 2: Furuta pendulum and notations of the different parameters.

Let τ_ϕ and τ_θ denote the external torques applied to the vertical arm and to the horizontal arm, respectively. Introducing

Name of parameter	Value	Unit	Notation
Angle of swinging link		rad	θ
Angle of horizontal link		rad	ϕ
Mass of horizontal link	0.072	kg	m_a
Mass of vertical link	0.00775	kg	m_p
Mass of the weight	0.02025	kg	M
Length of horizontal link	0.25	m	l_a
Length of vertical link	0.4125	m	l_p
Coulomb friction	0.015	Nm	τ_S
Coulomb stiction	0.01	Nm	τ_C
Maximal rotation speed for both links	2	$\frac{\text{rotation}}{s}$	
Approx. zero angular speed for swinging link	0.02	$\frac{rad}{s}$	$\dot{\phi}_\epsilon$
Time intervals between interrogations	100	ms	
Maximum control value	0.05	Nm	δ

Table 1: Parameters of the Physical Model

$$\begin{aligned}\alpha &= J + (M + \frac{1}{3}m_a + m_p)l_a^2, & (14)\\ \beta &= (M + \frac{1}{3}m_p)l_p^2, & (15)\\ \gamma &= (M + \frac{1}{2}m_p)l_a l_p, & (16)\\ \delta &= (M + \frac{1}{2}m_p)g l_p & (17)\end{aligned}$$

and using the external torques, we can write the equations of the dynamics of the pendulum as follows [5]:

$$\begin{aligned}(\alpha + \beta sin^2\theta)\ddot{\phi} + \gamma cos\theta\,\ddot{\theta} + 2\beta cos\theta\, sin\theta\,\dot{\phi}\,\dot{\theta} - \gamma sin\theta\,\dot{\theta}^2 &= \tau_\phi & (18)\\ \gamma cos\theta\,\ddot{\phi} + \beta\ddot{\theta} - \beta cos\theta\, sin\theta\,\dot{\phi}^2 - \delta sin\theta &= \tau_\theta & (19)\end{aligned}$$

The real pendulum exhibits significant friction in the $\phi-$joint. The friction can be modelled in several ways. We used Coulomb friction with stiction [5]:

$$\tau_F = \begin{cases} \tau_C\, sgn\dot{\phi} & \text{if } \dot{\phi} \neq 0, \\ \tau_u & \text{if } \dot{\phi} = 0 \text{ and } \|\tau_u\| \leq \tau_S, \\ \tau_S\, sgn\dot{\tau}_u & \text{otherwise} \end{cases} \qquad (20)$$

In our simulations the zero condition on the velocity is replaced by $\|\dot{\phi}\| \leq \dot{\phi}_\epsilon$, with $\dot{\phi}_\epsilon$ chosen according to [5].

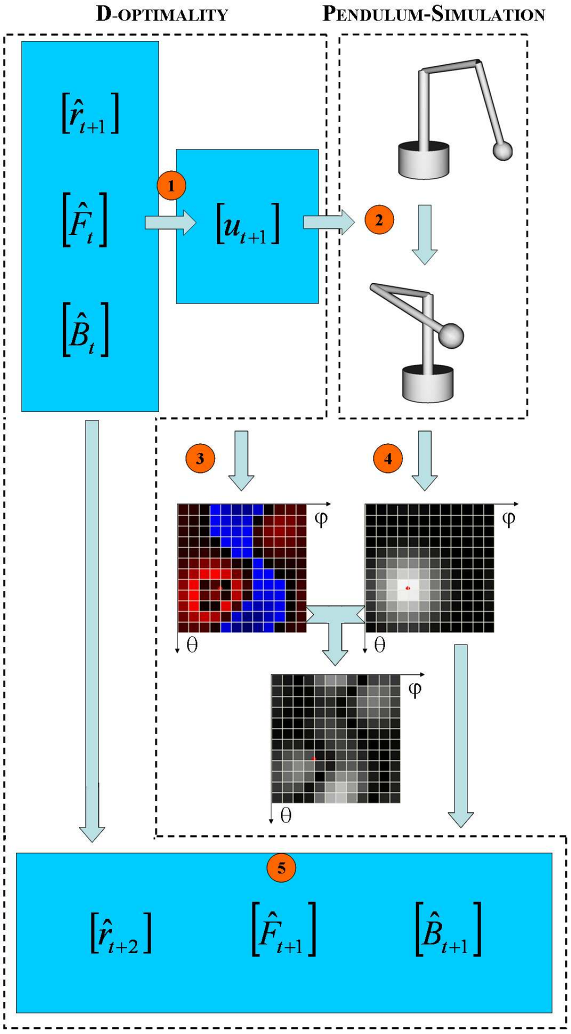


Figure 3: Scheme of D-optimal interrogation. (1) Control u_{t+1} is computed from D-optimal principle, (2) control acts upon the pendulum, (3) signals predicted before control step, (4) sensory information after control step. Difference between (3) and (4) is used for the computation of the cumulated prediction error. (5) Parameters were updated according to Algorithm 1. For more details, see text.

3.2 Simulation and learning

The pendulum is a continuous dynamical system that we observe in discrete time steps. Furthermore, we assume that our observations are limited; we have only 144 crude sensors for observing angles ϕ and θ. In each time step these sensors form our $r_t \in \mathbb{R}^{144}$ observations, which were simulated as follows: Space of angles ϕ and θ is $[0, 2\pi) \times [0, 2\pi)$. We divided this space into $12 \times 12 = 144$ squared domains of equal sizes. We 'put' a Gaussian sensor at the centre of each domain. Each sensor gives maximal response 1 when angles θ and ϕ of the pendulum are in the centre of the respective sensor, whereas the response decreased according to the Gaussian function. For all sensors, response r_i scaled as $r_i = \frac{1}{2\pi\sigma^2}\exp(-\frac{(\theta-\theta_i)^2+(\phi-\phi_i)^2}{2\sigma^2})$ $(1 \leq i \leq 144)$, where angles θ_i, ϕ_i correspond to the middle point of our 12×12 grid, and σ was set to 1.58 in radians. Sensors were crude but noise-free; no noise was added to the sensory outputs. The inset at label 4 of Figure 3 shows the outputs of the sensors in a typical case. Sensors satisfied periodic boundary conditions; if sensor S was centred around zero degree in any of the directions, then it sensed both small (around 0 radian) and large (around 2π radian) angles. We note that the outputs of the 144 domains are arranged for purposes of visualisation; the underlying topography of the sensors is hidden for the learning algorithm.

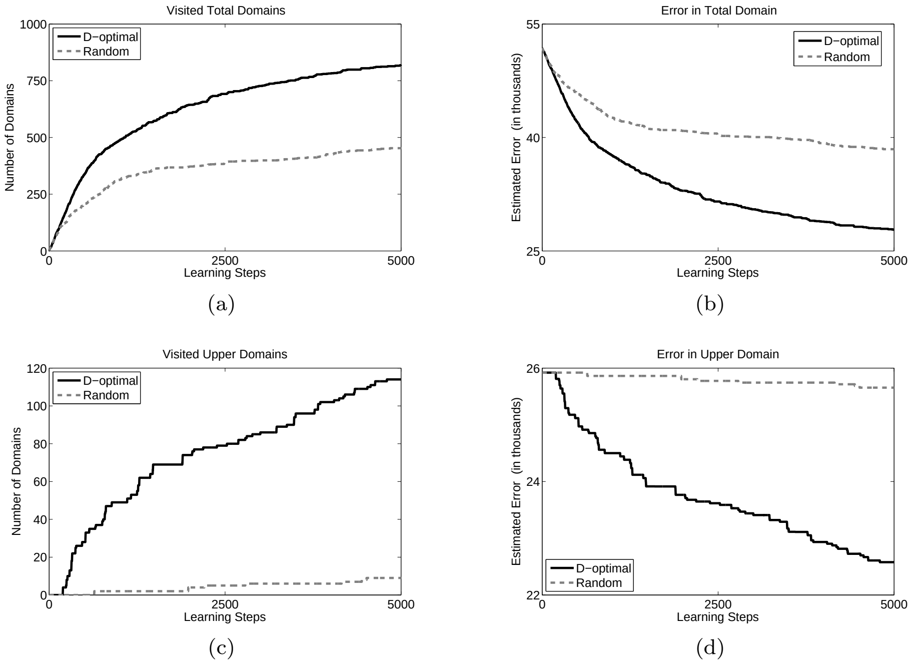
We observed these $r_t = (r_1(t), \ldots, r_i(t), \ldots, r_{144}(t))^T$ quantities and then calculated the $u_{t+1} \in \mathbb{R}^2$ D-optimal control using Algorithm 1, where we approximated the pendulum with the model $\tilde{r}_{t+1} = Fr_t + Bu_{t+1}$, $F \in \mathbb{R}^{144\times 144}$, $B \in \mathbb{R}^{144\times 2}$. Components of vector u_{t+1} controlled the two actuators of the angles separately. Maximal magnitude of each control signal was set to 0.05 Nm. Clearly we do not know the best parameters for F and B in this case, so in the performance measure we have to rely on prediction error and the number of visited domains. This procedure is detailed below.

First, we note that the angle of the swinging link and the angular speeds are important from the point of view of the prediction of the dynamics, whereas the angle of the horizontal link can be neglected. Thus, for the investigation of performance of the learning process, we used the 3D space determined by $\dot{\phi}, \theta$ and $\dot{\theta}$. As was mentioned above, angular speeds were restricted to the $[-2\frac{\text{rot}}{\text{s}}, 2\frac{\text{rot}}{\text{s}}]$ domain. We divided each angular speed domain into 12 equal regions. We also used the 12-fold division of angle θ. Counting the domains, we had $12 \times 12 \times 12 = 1,728$ rectangular block shaped domains. Our algorithm provides estimations for $\hat{F}_t$ and $\hat{B}_t$ in each instant. We can use them to compute the predicted observation vector $\hat{r}_{t+1} = \hat{F}_t r_t + \hat{B}_t u_{t+1}$. An example is shown in the inset with label 4 in Figure 3. We computed the absolute value of the prediction errors $e_i(t+1) = \|r_{i,t+1} - \hat{r}_{i,t+1}\|$ for all i, and cumulated them over all domains $(i = 1, \ldots 1,728)$ as follows. For each domain, we set the initial error value at 30, a value somewhat larger than the maximal error we found in the computer runs. Therefore the cumulated error at start was $1,728 \times 30 = 51,840$ and we measured how the error decreases.

4 Results

The D-optimal algorithm does two things simultaneously: (i) it explores new domains, and (ii) it decreases the errors in the domains already visited. Thus, we measured the cumulated prediction errors during learning and corrected the estimation at each step. So, if our cumulated error estimation at time t was $e(t) = \sum_{k=1}^{1,728} e_k(t)$ and the pendulum entered the i^{th} domain at time $t+1$, then we set $e_k(t+1) = e_k(t)$ for all $k \neq i$ and $e_i(t+1) = \|r_{i,t+1} - \hat{r}_{i,t+1}\|$. Then we computed the new cumulated prediction error, i.e., $e(t+1) = \sum_{i=1}^{1,728} e_i(t+1)$.

We compared the random and the D-optimality interrogation schemes. We show two sets of figures, Figures 4a and 4b, as well as Figures 4c and 4d. The upper set depicts the results for the full set of the 1,728 domains. It is hard for the random control to enter the upper domain by chance, so we also investigated how the D-optimal control performs here. We computed the performance for cases when the swinging link was above vertical, that is for 864 domains (Figs. 4c and 4d).

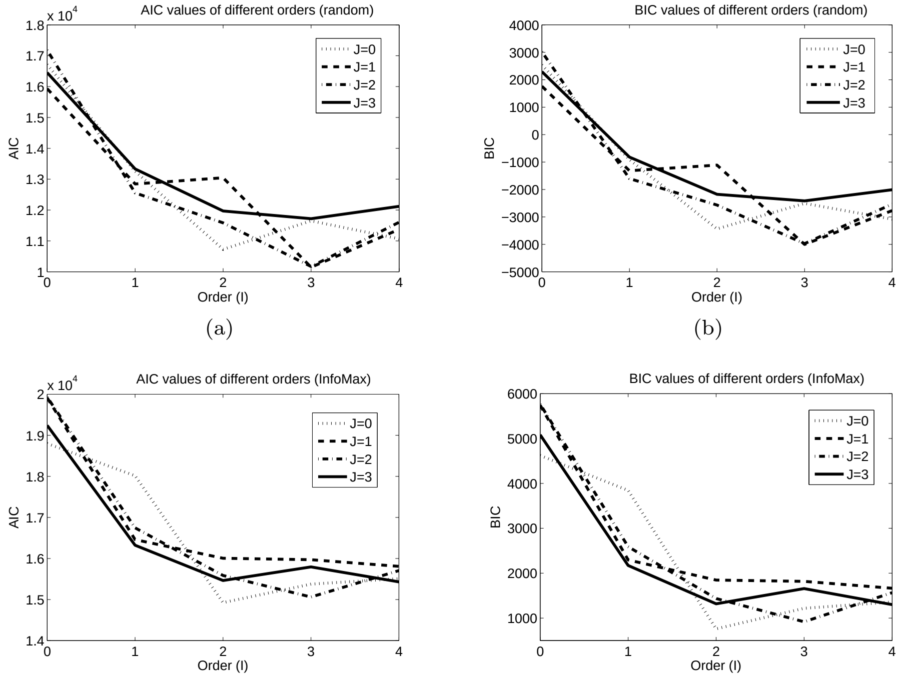


(a) (b)

(c) (d)

Figure 4: Furuta experiments driven by random and D-optimality controls. Solid (dotted) line: D-optimal (random) case. (a-b): Number of domains is 1728. (a): visited domains, (b): upper bound for cumulated estimation error in all domains, (c-d): Upper half of the space, i.e., the swinging angle is above horizontal and the number of domains is 864. (c): number of visited domains, (d): upper bound for cumulated estimation error. For more details, see text.

For the full space, the number of visited domains is 456 (26%) and 818 (47%) for the random control and the D-optimal control, respectively after 5,000 control steps (Fig. 4a). The error drops by 13,390 (26%) and by 24,040 (46%), respectively (Fig. 4b). While the D-optimal controlled pendulum visited more domains and



(a) (b)

(c) (d)

Figure 5: The Akaike's information criterion (AIC) and the Bayesian information criterion (BIC) values for random and InfoMax controls and for models up to fourth order ($I = 0, \ldots, 3$) and for different control orders ($J = 0, \ldots, 3$).

achieved smaller errors, the domain-wise estimation error is about the same for the domains visited; both methods gained about 29.4 points per domains.

We can compute the same quantities for the upper domains as well. The number of visited upper domains is 9 and 114 for the random control and for the D-optimal control, respectively (Figure 4c). The decrease of error is 265 and 3,342, respectively (Figure 4d). In other words, D-optimal control gained 29.3 points in each domain on average, whereas random control, on average, gained 29.4 points, which are very close to the previous values in both cases. That is, infomax control gains more information concerning the system to be identified by visiting new domains.

This observation is further emphasised by the following data: The infomax algorithm discovered 37 new domains in the last 500 steps of the 5,000 step experiment. Out of these 37 domains, 20 (17) were discovered in the lower (upper) domain. By contrast, the random algorithm discovered 9 domains, out which 5 (4) was in the lower (upper) domain. That is, infomax has a similar (roughly fourfold) lead in both the upper and lower domains, although the complexity of the task is different and the relative number of available volumes is also different in these two domains.

We studied the learning process as a function of the Gaussian width. We found that learning is robust in this respect: the estimation error was a very weak function of σ, the spread of the Gaussian, except for very small variance Gaussians. Learning was spoiled for very broad Gaussians, too.

By construction, there is a second order dynamical system in the background, so we studied if one can find this order as the result of the learning process. We calculated Akaike's information criterion (AIC) and the Bayesian information criterion (BIC) values of the model for different control orders. Figure 5 shows the AIC and BIC values of both random control and for InfoMax control. We measured the values for models up to fourth order ($I = 0, \dots, 3$), for control orders $J = 1, \dots, 3$, and for $\sigma = 1.58$ radian. There is a large gain if one increases $I = 0$ to $I = 1$, i.e., if one assumes second order dynamics. Further increases of I give smaller improvements. The only exception is the case of $J = 0$. Here, the improvement is not so sudden between $I = 0$ and $I = 1$ and considerable further drops can be seen for $I = 2$. This is most prominent under the InfoMax conditions. Thus, there is a memory effect in the control: control rendered by InfoMax at time t may depend on the control rendered by InfoMax at time $t-1$. This dependency is learned and is represented by matrix F_2. From the point of view of the order of the control, there is little dependence here, except for the case of $I = 1$: there is a large performance difference – for InfoMax control – between $J = 0$ and $J = 1$. Again, this points to the memory effect in InfoMax, which can be uncovered by matrix B_1. Taken together, the approach can provide an *estimation about the order of the dynamical system*, but *not* in the InfoMax operation mode.

Finally, we note that the InfoMax procedure, which we demonstrated here on the case of the Furuta pendulum, may gain from discovering the direct product space behind the 144 sensors. Then explorations might concern the low-dimensional direct product space, instead of the raw sensory observations.

5 Summary and conclusion

In this paper we have studied the InfoMax learning for the two-link Furuta pendulum. We used a slightly modified version of the generalised linear model described in [8]. The intriguing property of this slight modification is that it leads to strikingly simple learning rules [11].

InfoMax intends to optimise the next control action given what has been learned and what has been observed. We demonstrated that this online (i.e., real time) learning method explores larger areas than random control without significant compromise in the precision of the estimation in the visited domains. The discovery rate is in favour of the InfoMax algorithm, which had similar leads in the domains which were easier to find and in the domains, which were harder to find.

The pendulum problem also shows the limitations of the InfoMax solution. This is a low-dimensional problem and InfoMax cannot learn the hidden regularities. Connections to reinforcement learning should be established for efficient learning. Convergent methods that can connect InfoMax learning and reinforcement learning seem important for machine learning.

References

[1] Boyen, X. and Koller, D. Tractable inference for complex stochastic processes. In *Fourteenth Conference on Uncertainty in Artificial Intelligence*, pages 33–42, 1998.

[2] Cohn, D. A. Neural network exploration using optimal experiment design. In *Advances in Neural Information Processing Systems*, volume 6, pages 679–686, 1994.

[3] Fedorov, V. V. *Theory of Optimal Experiments.* Academic Press, New York, 1972.

[4] Furuta, K., Yamakita, M., and Kobayashi, S. Swing-up control of inverted pendulum using pseudo-state feedback. *Journal of Systems and Control Engineering*, 206:263–269., 1992.

[5] Gäfvert, M. Modelling the furuta pendulum. Technical report ISRN LUTFD2/TFRT–7574–SE, Department of Automatic Control, Lund University, Sweden, April 1998.

[6] Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. *Bayesian Data Analysis.* CRC Press, 2nd erdition, 2003.

[7] Gupta, A. K. and Nagar, D. K. *Matrix Variate Distributions*, volume 104 of *Monographs and Surveys in Pure and Applied Mathematics.* Chapman and Hall/CRC, 1999.

[8] Lewi, J., Butera, R., and Paninski, L. Real-time adaptive information-theoretic optimization of neurophysiology experiments. In *Advances in Neural Information Processing Systems*, volume 19, 2007.

[9] Minka, T. *A family of algorithms for approximate Bayesian inference.* PhD thesis, MIT Media Lab, MIT, 2001.

[10] Opper, M. and Winther, O. A Bayesian approach to online learning. In *Online Learning in Neural Networks.* Cambridge University Press, 1999.

[11] Póczos, B. and Lőrincz, A. D-optimal Bayesian interrogation for parameter estimation and noise identification of recurrent neural networks. Technical Report, Janury 2008. `http://uk.arxiv.org/PS_cache/arxiv/pdf/0801/0801.1883v1.pdf`.

[12] Solvang, B., Korondi, P., Sziebig, G, and Ando, N. SAPIR: Supervised and adaptive programming of industrial robots. In *11th IEEE International Conference on Intelligent Engineering Systems*, INES07, Budapest, Hungary, June 2007.

[13] Szepesvári, Cs., Cimmer, Sz., and Lőrincz, A. Neurocontroller using dynamic state feedback for compensatory control. *Neural Networks*, 10:1691–1708, 1997.

[14] Szepesvári, Cs. and Lőrincz, A. Approximate inverse-dynamics based robust control using static and dynamic feedback. *Applications of Neural Adaptive Control Theory II*, 2:151–179, 1997. World Scientific, Singapore.