Four Encounters with System Identification

Lennart Ljung, Håkan Hjalmarsson and Henrik Ohlsson
Div. Automatic Control, Linköping University, Sweden, ljung,ohlsson@isy.liu.se
The Automatic Control LAB, Royal Institute of Technology, Stockholm, Sweden, hjalmars@kth.se

***Abstract*— Model-based engineering becomes more and more important in industrial practice. System identification is a vital technology for producing the necessary models, and has been an active area of research and applications in the automatic control community during half a century. At the same time, increasing demands require the area to constantly develop and sharpen its tools. This paper deals with how system identification does that by amalgamating concepts, features and methods from other fields. It describes encounters with four areas in systems theory and engineering: Networked Systems, Particle Filtering Techniques, Sparsity and Compressed Sensing, and Machine Learning. The impacts on System Identification methodology by these encounters are described and illustrated.**

I. Introduction

System Identification is the art and science of building mathematical models of dynamical systems from observed input- output signals. It is a rather old and mature field with roots in automatic control, at least from 1956, [98] and with basic techniques going back several centuries to Gauss, [28]. Nevertheless the topic remains vital and vibrant, not the least because demands from model-based engineering require constant development and improvement of tools. The sustained scientific interest in System Identification is evidenced, e.g. by the invitation to the current plenary, the organization of many conferences/conference sessions, and the interest in publications and software in the area. The authors of the current paper also just received a five-year advanced research grant from the European Research Council for studies in "Limitations, Estimation, Adaptivity, Reinforcement, Networks (LEARN) in System Identification."

It is the purpose of the current paper to give a background and some details of what keeps System Identification alive and kicking. We will do that by telling about four encounters where system identification meets and tries to absorb the essence of new techniques for pushing the identification methodology forward. It must be stressed that the four areas are just examples of the development of the identification field. Other authors could and would have made other selections in dealing with the essential progress of the field.

II. System Identification Meets Networked Systems

A. Introduction

The profound importance that networked systems play in our lives today is of course evidenced by the Internet. Perhaps less obvious is an on-going "hidden" network revolution in a number of technology areas, e.g., in automotive engineering, where networks are replacing expensive traditional wiring. Cheap computational and wireless transmission devices combined with a wide range of emerging new sensors and actuators, in turn enabled by new technologies such as micro- and nanosystems, have opened up for the use of networked control systems on a large scale. The control community has been quick in catching up to address the associated theoretical challenges and in recent years this has been a vibrant and exciting research area [4].

Characteristic to networked systems are that they consist of "components" that are spatially distributed. Each component can be viewed as consisting of a physical part and an engineered device, in turn consisting of sensors, actuators and a processing unit. The physical parts may be interconnected in rather arbitrary ways (depending on the type of system) while the processing units are interconnected by communication networks, see Figure 1.

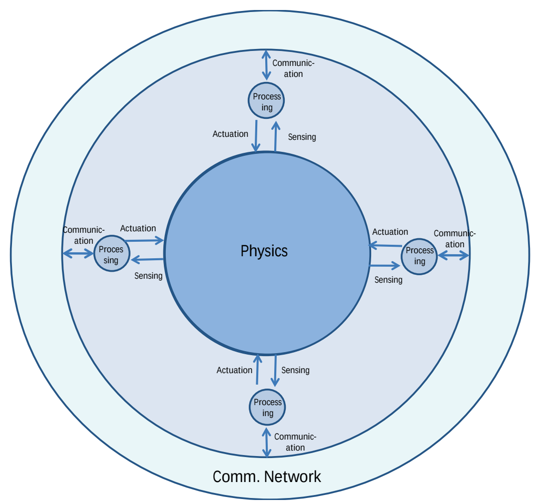

Fig. 1. A networked system.

This configuration changes the perspective of control design in two aspects: Firstly, the (dynamic) properties of the communication network have to be accounted for, and, secondly, the distributed nature of the system is emphasized. Large efforts have been (and still are) devoted to develop dynamic models for communication networks. These models depend very much on the particular technology that is used, e.g. wired and wireless networks exhibit widely different characteristics. In particular, attention has been given to wireless networks, but there is also extensive work on wired networks, e.g. the Internet [75]. Some features of wireless networks include sampling time jitter, random packet losses and delays, and requirements of low grade quantization. Constraints on power consumption, costs, and channel capacity are consequences of the distributed nature which limits local information processing and information exchange between the components.

System identification problems related to networked systems can broadly be divided into two categories:

1) How to identify models of the communication network itself.
2) How to identify models of the physical system (possibly including sensor and actuator dynamics) in a networked/distributed environment.

In this section we will in very select manner – our coverage is by no means complete – highlight some system identification problems that relate to these two problems.

B. Identification of communication networks

We will illustrate one important aspect of identification of communication networks by discussing congestion control of internet traffic. In the seminal paper [43] Frank Kelly and coworkers presented a framework for the analysis and synthesis of congestion control of internet traffic. The data traffic is aggregated into fluid flows and by interpreting the indirect signaling that takes place in the network, e.g. queuing delays, as prices, the congestion control problem can be solved as a decentralized convex optimization program.

Since Kelly's work, the underlying fluid flow model has undergone extensive refinement, see, e.g., [50], [73], [51], [85], [40], [75]. In Figure 2 a generic communication network is depicted. The network is used by N sources, corresponding to N persistent flows in the network. Source n sends $x_n(t)$ packets per second at time t into the network. The signal x in the network represents a vector with the rates of all the sources. The network consists of L links, with associated finite capacities c_l, $l = 1, \ldots, L$ (in packets per second). The interconnection structure can be defined via a so called *routing matrix* $R \in \mathbb{R}^{L\times N}$ for which element (l, n) is 1 if link l is used by source n, and 0 otherwise. The signal y in the figure represents a vector with the aggregate flows $y_l(t)$ for all links $l = 1, \ldots, L$. The link flows are given by

$$y_l(t) = \sum_{n=1}^{N} R_{ln} x_n(t - \tau_{ln}^f) =: r_f(x(t), \tau_l^f)$$

where τ_{ln}^f is the time it takes for a packet that is sent by source n to reach link l. Link l reacts to its level of congestion by responding with a "price" signal p_l. The price signal depends on the design of the network. One possibility is to let p_l be a function of the queuing time experienced at link l. In Figure 2, $\mathcal{L}$ represents these link dynamics. The prices (collected in the vector p) are sent back to the sources upon which source n receives the aggregate price

$$q_n(t) = \sum_{l=1}^{L} R_{ln} p_l(t - \tau_{ln}^b) =: r_b(p(t), \tau_n^b). \qquad (1)$$

where τ_{ln}^b is the time it takes for the price at link l to reach source n. The block H in Figure 2 indicates that the aggregate prices may be distorted (e.g. quantized) before the sources get access to them. Depending on the experienced price, each source adjusts it transmit rate. This source dynamics is represented by $\mathcal{S}$ in the figure.

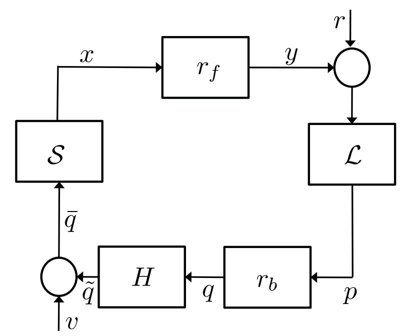

Fig. 2. Schematic representation of the congestion control system.

A fluid flow model only includes persistent sources. Intermittent traffic can be modeled as variations in the link rates and is represented by the signal r in Figure 2. Furthermore, the model only includes bottleneck links, i.e. those links l operating at their maximum capacity c_l. The signal v models variations in the received aggregated prices due to that non-bottleneck links are exposed to short term traffic.

Several studies have been conducted to validate the model above, e.g. [90] and [85]. In particular, interest has focused on identifying the link dynamics. Notice that congestion control systems are feedback systems, c.f. Figure 2, and that care therefore has to be exercised when data from such systems are used for identifying or validating a model. To illustrate what can happen, suppose that we would like to identify the link dynamics from measurements of source rates x and link prices p. Suppose first that only r is excited. Then it is easy to see that

$$x = \mathcal{S} H r_b^T p$$

from which we see that identifying a model

$$p = Gx \qquad (2)$$

will result in that the inverse of source dynamics $\mathcal{S}$ is identified, rather than the desired link dynamics. This is exactly the well known problem that the inverse of the controller is identified under certain excitation conditions [47]. The proper experiment should be carried out with excitation in v since then

$$p = \mathcal{L} r_f x$$

implying that the link dynamics is identified with the model (2). In practice variations in v can be obtained by manipulating the protocol at the source which determines the source dynamics, e.g., the TCP protocol.

As most communication systems operate in closed loop, the discussion above shows that system identification certainly can contribute to identification of communication networks. For more details on how to identify congestion control dynamics in a proper manner we refer to [39].

C. Decentralized identification in a networked environment

We will now consider some aspects that arise when a system is to be identified in a networked environment characterized by limitations in data communication and the possibility/necessity of local data processing, e.g., due to the communication constraints. Two essential scenarios can be considered:

1) Fusion centric
2) Fully decentralized

where in the first scenario nodes transmit information to a fusion center for final processing, whereas in the second scenario no such center exists but nodes have to update each other with as little coordination, synchronization and communication as possible. Next we will discuss the statistical basis for identification under such schemes.

1) A statistical basis: Let y be a random vector with probability density function (pdf) $p_\theta(\cdot)$ where $\theta \in \mathbb{R}^n$. The Cramér-Rao Lower Bound (CRLB) provides a lower bound for any unbiased estimator $\hat{\theta}$ of θ that is based on y. Subject to certain regularity conditions (see [44]), the covariance matrix of the parameter estimate is lower bounded by

$$\mathrm{E}\left[(\hat{\theta}-\theta)(\hat{\theta}-\theta)^T\right] \geq I_F^{-1}(p_\theta) \tag{3}$$

where $I_F(p_\theta)$ is the Fisher information matrix

$$I_F(p_\theta) = -\mathrm{E}\left[\frac{\partial^2}{\partial\theta\partial\theta^T}\log p_\theta(y)\right] \tag{4}$$

Under certain conditions the CRLB is achieved asymptotically (as the number of measurements grows) by the maximum likelihood (ML) estimate:

$$\hat{\theta}^{ML} = \arg\max_\theta p_\theta(y) \tag{5}$$

Consider now the problem of estimating θ in a networked system. Let p_θ^{tot} denote the joint pdf of all measurements in the system. Then $I_F^{-1}(p_\theta^{tot})$ is a universal lower bound for the covariance matrix of any unbiased parameter estimate. Notice that (5) requires the processing of all measurements, i.e. all data has to be gathered at a "fusion center" and be processed there, according to (5). When measurements are processed locally before being transmitted to the fusion center one may therefore expect a loss in accuracy. However, this is not generically true and to understand when local processing is possible without loss of accuracy we will need the concept of a sufficient statistic. We say that $s = s(y)$ is a sufficient statistic for y if the conditional distribution of y given s, $p(y|s)$ say, is independent of θ [44]. This means that given s it is possible to generate random samples from $y|s$ without knowing θ and hence to generate a new random vector $\tilde{y}$ which has exactly the same distribution as y.

Example 1: Suppose that the elements of $y = \left[y_1, \ldots, y_m\right]^T \in \mathbb{R}^m$ are independently normal distributed with unknown mean $\theta \in \mathbb{R}$ and known variance λ. Then the sample mean of y is a sufficient statistic. The sample mean is also the ML estimate $\hat{\theta}^{ML}$ of θ.

If in addition λ is unknown, the sample mean and the sample variance of the residuals

$$\hat{\lambda} = \frac{1}{m}\sum_{j=1}^{m}(y_j - \hat{\theta}^{ML})^2$$

form a sufficient statistic. ■

There is no need to re-create "y" from a sufficient statistic by the random procedure outlined above, the sufficient statistic can be used directly in the estimation. This can be seen as follows. Let $\tilde{p}_\theta(s)$ be the pdf of s, then

$$\begin{aligned} I_F(p_\theta) &= -\mathrm{E}\left[\frac{\partial^2}{\partial\theta\partial\theta^T}\log p_\theta(y)\right] \\ &= \{p_\theta(y) = p(y|s(y))\tilde{p}_\theta(s(y)), \\ &\quad \text{(by assumption } p(y|s) \text{ does not depend on } \theta)\} \\ &= -\mathrm{E}\left[\frac{\partial^2}{\partial\theta\partial\theta^T}\log \tilde{p}_\theta(s(y))\right] = I_F(\tilde{p}_\theta) \end{aligned}$$

implying that the CRLB's using y and s are identical since the information matrices are equal.

From a statistical point of view there is no information loss if the local processing of measurements generates sufficient statistics. We illustrate this with an example.

Example 2: Suppose that there are two sensors, where Sensor i) provides the measurements $y_i = \left[y_{i1}, \ldots, y_{im}\right]^T \in \mathbb{R}^m$ according to

$$y_{ij} = \theta + e_{ij}, \quad j = 1, \ldots, m$$

where $\{e_{ij}\}$ are independent normally distributed random variables with zero mean.

The local ML estimate of θ based on the measurements from Sensor i) only are given by

$$\hat{\theta}_i^{ML} = \frac{1}{m}\sum_{j=1}^{m} y_{ij}$$

These estimates can be computed locally and then transmitted to a fusion center where, e.g., the estimate

$$\hat{\theta}^{central} = \frac{\hat{\theta}_1^{ML} + \hat{\theta}_2^{ML}}{2} \tag{6}$$

can be formed. If the variance of the measurment errors from the sensors are equal, this is the ML estimate given the measurements from both sensors. However, when the measurement errors are different, i.e. when $\mathrm{E}[e_{ij}^2] = \lambda_i$, $\lambda_1 \neq \lambda_2$, (6) is no longer the central ML estimate. Suppose that Sensor 1 is of high quality such that λ_1 is small but that Sensor 2 is very poor so that $\lambda_2 \gg \lambda_1$. Then $\hat{\theta}^{central}$ will in fact be a worse estimate than $\hat{\theta}_1^{ML}$.

A better estimate at the fusion center can be obtained by transmitting the local sufficient statistics s_i instead of the local ML estimates. Let $y = \left[y_1^T, y_2^T\right]^T \in \mathbb{R}^{2n}$ and let s_i be a sufficient statistic for y_i. Since the sensors are subject to independent noise it holds

$$p_\theta(y) = \prod_{i=1}^{2} p_\theta(y_i) = \prod_{i=1}^{2} p(y_i|s_i) \prod_{i=1}^{2} p_\theta(s_i) \qquad (7)$$

which shows that $s = \left[s_1^T, s_2^T\right]^T$ is a sufficient statistic for the total measurement vector y. From Example 1 we see that if we in addition to the local ML estimates should transmit also the local noise variance estimates

$$\hat{\lambda}_i = \frac{1}{m}\sum_{j=1}^{m}(y_i j - \hat{\theta}_i^{ML})^2$$

These variance estimates can then be used by the fusion center to avoid the problem that a single poor sensor may destroy the fused estimate of θ. ■

In conclusion, regardless of fusion centric or fully decentralized schemes, unless sufficient statistics are transmitted information loss will occur.

A subtle issue arises when the nodes are privy to local information regarding their own measurement process. We discuss this through an example.

Example 3: Suppose that we have n nodes and that node i measures $y_i = \begin{bmatrix} y_{i1} & \dots & y_{im}\end{bmatrix}^T \in \mathbb{R}^m$ where each element is independently normal distributed with unknown mean θ and variance λ_i. Let us assume that observations at different nodes are independent. Then the central ML estimate of θ is given by

$$\hat{\theta}^{ML} = \frac{\sum_{i=1}^{n} \frac{\bar{y}_i}{\lambda_i}}{\sum_{i=1}^{n} \frac{1}{\lambda_i}} \qquad (8)$$

where $\bar{y}_i$ is the sample mean at node i

$$\bar{y}_i = \frac{1}{m}\sum_{t=1}^{m} y_{ij}$$

Suppose now that each node knows its own noise variance but not the others'. Then, following Example 1, $\bar{y}_i$ is a sufficient statistic. However, in order to combine the local sufficient statistics in the right way, i.e. as in (8), also the noise variances λ_i have to be communicated. ■

There exist more or less elaborate ways to deal with the problem highlighted in Example 3. In a fully decentralized context [95] communicates the necessary weights explicitly whereas in [5] the weighting is done locally.

Another subtle issue arises when the transmission of information from the sensors to the fusion center is subject to rate constraints. It may then happen that *ancillary* statistics may improve upon the estimate. An ancillary statistic is a statistic whose distribution does not depend on the unknown parameters. We refer to the surveys [34], [94] and references therein for details of the problem when the transmission to the fusion center is rate constrained.

2) The impact of noise: In Example (2) it is the fact that the nodes have access to independent measurements, i.e. (7), that ensures that it is sufficient to distribute the local sufficient statistics between nodes, or to a fusion center. When this is not the case, the situation is much more complex and in this section we will illustrate that the spatial correlation properties of the noise are crucial for how data should be processed. There exists a simple condition for when locally linearly processed measurements can be combined into the centralized linear minimum mean variance estimate [71].

Consider the distributed system in Figure 3. Node i has y_{i-1} as input and y_i as output and consists of the first order transfer function $G_i(q) = b_i q^{-1}$, where the parameters b_i, $i = 1, \dots, n$ need to be estimated. We denote estimates by $\hat{b}_i$ and the true values by b_i^o. Suppose that, for some reason, the interest is to estimate $J_o = \sum_{k=2}^{n} b_k^o$.

We will consider three different estimation schemes: A fusion centric approach where $\{y_{i-1}(t), y_i(t)\}_{t=1}^N$ is used locally to estimate b_i using least-squares estimation, which then is sent to a fusion center where all estimates are combined. The second scheme will be a centralized scheme where all parameters are estimated jointly using $\{y_i(t)\}_{t=1}^N$, $i = 0, \dots, n$. In the last approach neighboring nodes collaborate by passing data between each other before subsequent data processing and transmission to the fusion center.

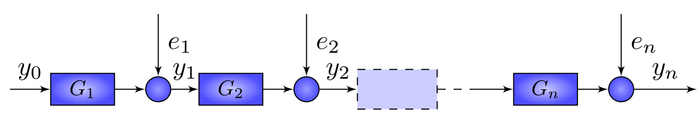

Fig. 3. A decentralized system.

We will assume that for $i = 1, \dots, n$, $\{e_i(t)\}_{t=1}^N$ is a sequence of independent normal distributed random variables with zero mean and known variance λ. In regards to the spatial properties of the noise we will consider two extreme cases: Firstly we will assume that the noise sequences are mutually independent. In the second scenario the noise sequences are perfectly dependent, i.e. $e_i(t) = e(t)$, $i = 1, \dots, n$, $t = 1, \dots, N$ where $\{e(t)\}_{t=1}^N$ is a sequence of independent random variables. That disturbances acting on various subsystems can be strongly correlated is not an uncommon situation. Consider for example an experiment on a power grid during a thunderstorm with lightnings.

In order to simplify the calculations we assume that $b_i^o = 0$, $i = 1, 2, \dots, n$, so that

$$y_i(t) = e_i(t) \qquad (9)$$

and that the first input y_0 is zero mean white noise, also with variance λ. We consider first the case where the noise sequences are mutually independent.

Independent noise sequences. When $\{e_i(t)\}_{t=1}^N$ is independent from all other noise sequences as well as normally distributed, the ML estimate of b_i is obtained as the least-squares estimate corresponding to

$$y_i(t) = b_i y_i(t-1) + e_i(t), \quad t = 1, \ldots, N \tag{10}$$

i.e.

$$\hat{b}_i = \frac{\sum_{t=1}^{N} y_i(t) y_i(t-1)}{\sum_{t=1}^{N} y_i^2(t-1)} \tag{11}$$

Thus the fusion centric scheme gives exactly the same result as the centralized scheme in the case of mutually independent noise sequences. We also conclude that there is nothing to gain by allowing the nodes to share information during the data processing.

Dependent noise sequences. When the noise sequences are identical, we obtain from (9) and (11) that the local least-squares estimates are given by

$$\begin{aligned}\hat{b}_i &= b_i^o + \frac{\frac{1}{N}\sum_{t=1}^{N} e(t) y_i(t-1)}{\frac{1}{N}\sum_{t=1}^{N} y_i^2(t-1)} \\ &= b_i^0 + \bar{e}, \ i = 2, \ldots, n \end{aligned} \tag{12}$$

where $\bar{e}$ is an error term

$$\bar{e} = \frac{\frac{1}{N}\sum_{t=1}^{N} e(t) e(t-1)}{\frac{1}{N}\sum_{t=1}^{N} e^2(t-1)} \tag{13}$$

that is common to all estimates. From (12) we obtain that for large N (so that $\frac{1}{N}\sum_{t=1}^{N} e^2(t-1) \approx \lambda$),

$$\mathrm{E}[(\hat{b}_i - b_i^o)^2] \approx \frac{\frac{1}{N^2} N\lambda^2}{\lambda^2} = \frac{1}{N} \tag{14}$$

but also, since $\bar{e}$ is common to all estimates,

$$\mathrm{E}[(\hat{b}_i - b_i^o)(\hat{b}_j - b_j^o)] \approx \frac{1}{N}, \quad i = 2, \ldots, n$$

This implies that the mean-squared error of the estimate $\hat{J} = \sum_{k=2}^{n} \hat{b}_k$ is given by

$$\mathrm{E}[(\hat{J} - J_o)^2] \approx \frac{(n-1)^2}{N} \tag{15}$$

This means that for a large network (large n), even though the estimation error in each node is small, the total error can accrue to an unacceptable level.

In the centralized approach, when we can use measurements of all signals y_i, we can obtain perfect estimates of all parameters. Plugging in $e(t) = y_1(t) - G_1(q)u(t)$ into the equations for each node gives

$$\begin{aligned} y_i(t) &= G_i(q) y_{i-1}(t) + e(t) \\ &= G_i(q) y_{i-1}(t) + y_1(t) - G_1(q) y_0(t) \end{aligned}$$

which is a noise free relationship between variables, which means that the corresponding model parameters can be obtained exactly. Consequently also the estimate of J_o will be exact.

If now the nodes are allowed to communicate locally with each other, one can easily see that if node k sends a few samples of its input y_{k-1} to node $k+1$, then node $k+1$ can estimate b_{k+1} and b_k perfectly.

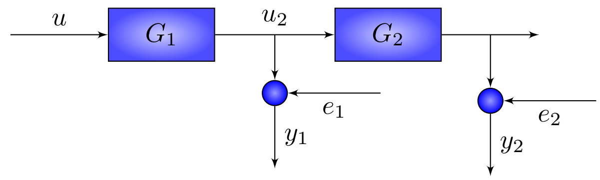

Fig. 4. A distributed multi-sensor network. The sensor measurements y_1 and y_2, together with the input u are used by a fusion center to estimate G_1 and G_2.

Summarizing, independent noise sequences leads to that fusion centric identification without collaboration between nodes is optimal, whereas the same scheme can have very poor performance when the noise sources are strongly correlated. In the latter case, local information exchange can significantly improve the accuracy. We conclude that the correlation structure of the noise can have a tremendous impact on how decentralized identification should be performed.

3) The impact of structure: Consider the distributed multi-sensor network in Figure 4 where sensor i may transmit the measurement y_i to a fusion center. Both transfer functions G_1 and G_2 are unknown. Consider now the identification of the transfer function G_1. From the figure it is clear that measurements from Sensor 1 provide information regarding this transfer function. However, it also seems somewhat obvious that Sensor 2 can contribute to improving the accuracy of the estimate of G_1 since $y_2 = G_2 G_1 u + e_2$. However, in [38], [83] it is shown that under certain configurations of the model structures used for G_1 and G_2, Sensor 2 contains almost no information regarding G_1 even if the signal to noise ratio of this sensor is very high. The result can easily be generalized to arbitrary number of nodes in a cascade structure.

4) Communication aspects: We will now discuss some of the issues associated with transmitting information over a communication channel.

a) Bandwidth limitations: One aspect is that the capacity of some communication channels is so low that it has to be accounted for. One such typical constraint is that the available bit rate may be restricted so that the information has to be quantized before transmission.

Example 4: Suppose that node i in a network measures the scalar

$$y_i = \theta + e_i$$

where e_i are independent normal distributed random variables with zero mean and variance λ. When each node only can relay whether y_i exceeds a certain threshold T (common to all nodes) or not, to the fusion center, it can be shown that the CRLB is at least a factor $\frac{\pi}{2}$ above the CRLB for the non-quantized case [62]. The optimal threshold is $T = \theta$, and therefore infeasible since θ is unknown. We refer to [67] for further details. ■

Results on quantized identification in a very general setting can be found in [86], [87], [88].

b) Sampling time jitter: Consider the scalar continuous time system

$$y(t) = \theta u(t) + v(t)$$

where $\theta \in \mathbb{R}$ is an unknown parameter to be identified. and where $v(t)$ is a disturbance. The input u, which we assume to be a stationary process with covariance function $r_u(\tau) = \mathrm{E}[u(t+\tau)u(t)]$, is uniformly sampled with sampling period T resulting in $u_n = u(nT)$. However, a non-ideal sensor causes sampling jitter τ_n in the corresponding output samples

$$y_n = \theta u(nT + \tau_n) + v_n \tag{16}$$

where $v_n = v(nT+\tau_n)$. Standard least-squares identification of θ using N samples from (16) gives

$$\begin{aligned}\hat{\theta} =& \frac{\frac{1}{N}\sum_{n=1}^{N} y_n u_n}{\frac{1}{N}\sum_{n=1}^{N} u_n^2} \\ =& \frac{\frac{1}{N}\sum_{n=1}^{N} u(nT+\tau_n)u(nT)}{\frac{1}{N}\sum_{n=1}^{N} u^2(nT)}\,\theta + \\ & \frac{\frac{1}{N}\sum_{n=1}^{N} v(nT+\tau_n)u(nT)}{\frac{1}{N}\sum_{n=1}^{N} u^2(nT)}\end{aligned} \tag{17}$$

Assuming that the jitter $\{\tau_n\}$ is stochastic with a stationary distribution and is independent of the noise, and that the noise has zero mean and is uncorrelated with the input, the second term on the right-hand side of (17) converges to zero as the number of samples $N \to \infty$. For the first term in (17) we have

$$\lim_{N\to\infty} \frac{\frac{1}{N}\sum_{n=1}^{N} u(nT+\tau_n)u(nT)}{\frac{1}{N}\sum_{n=1}^{N} u^2(nT)}\,\theta = \frac{\mathrm{E}_\tau[r_u(\tau)]}{r_u(0)}\,\theta \tag{18}$$

with the convergence being with probability 1 under suitable regularity conditions. In (18), $\mathrm{E}_\tau[r_u(\tau)]$ is the expectation of $r_u(\tau)$ with respect to the distribution of the jitter τ. Thus we see that the least-squares estimate of θ will not be consistent unless $\mathrm{E}_\tau[r_u(\tau)] = r_u(0)$. This illustrates that sampling jitter can cause bias problems in system identification. In [26], the jitter problem is analyzed in the frequency domain and it is shown how to compensate for the incurred bias. In the case of time-stamping of packets, a continuous time instrumental variable method is used in [84] to cope with irregular sampling.

D. Relation to other areas

There are obvious connections between fully distributed identification and distributed optimization, see e.g. [79], [9]. A widely studied area is distributed estimation where nodes share information regarding some random vector that is to be estimated. Relevant questions are whether the nodes will converge to the same estimate as the information exchange increases, and if so, whether the nodes reach a consensus and the quality of this estimate as compared to the centralized estimate, see, e.g., [12]. Wireless sensor networks is also a closely related area [52]. A popular class of methods are consensus algorithms where nodes are ensured to converge to the same result despite only local communication between nodes.

E. Summary

We have highlighted that it is important to take into account the closed loop nature of the problem when identifying communication networks. We have also seen that identification of networked systems is a multifaceted problem with close ties to fields such as distributed estimation and optimization. Whereas for a fix set-up, the CRLB provides a lower bound on the estimation accuracy, the main challenge when there are communication constraints is to devise the entire scheme: When, what and where should a node transmit in order to maximize accuracy? For example, suppose that it is possible to transmit all raw data to a fusion center but that the communication channel is subject to noise. Then it may still be better to pre-process the measurements locally before transmission. This type of considerations opens up a completely new ball-park for system identification. An interesting avenue is to view the problem as a decentralized optimization problem, e.g. using the methods in [13].

III. System Identification Meets Particle Filters

A. Identification of Nonlinear State Space Models

A general, discrete time, nonlinear identification model can be stated like this:

$$x(t+1) = f(x(t), u(t), v(t), \theta) \tag{19a}$$
$$y(t) = h(x(t), e(t), \theta) \tag{19b}$$

Here u and y are the inputs and the outputs of the nonlinear dynamical system, v and e are white noise disturbances (called process and measurement noises, resp.), θ is an unknown parameter vector. Often, in statistical literature f and h are determined in terms of the conditional transition probabilities from $x(t)$ to $x(t+1)$, $p_\theta(x(t+1)|x(t))$, and the conditional observation probabilities $q_\theta(y(t)|x(t))$. Also (19) is a *hidden Markov model*: the states x form a Markov process (due to the whiteness of v), and it is hidden, since only y is observed.

The parameter vector θ can generally be estimated by the *Maximum Likelihood* (ML) method. The negative log likelihood function (conditioned on the initial state $x(0)$, denoting past data by $Y^t = \{y(1), \ldots, y(t)\}$, can readily be written as

$$\hat{y}(t|\theta) = E(y(t)|Y^{t-1}, U^{t-1}) \tag{20a}$$
$$\varepsilon(t,\theta) = y(t) - \hat{y}(t|\theta) \tag{20b}$$
$$V(\theta, Y^N) = -\log P_\theta(Y^N) = \sum_{t=1}^{N} p_\nu(\varepsilon(t,\theta)) \tag{20c}$$

where $\hat{y}$ is the conditional expectation of the next output and p_ν is the probability density function (pdf) of the *innovations* ε. V is the log likelihood function.

The ML estimate of the parameters based on N data is then the parameter that minimizes the negative logarithm of the likelihood function:

$$\hat{\theta}(N) = \arg\min_{\theta} V(\theta, Y^N) \tag{20d}$$

B. Nonlinear Filtering and Particle Filtering (PF)

The catch here is the prediction $\hat{y}(t|\theta)$. In case the model (19) is linear and the noises are Gaussian, the predictor is readily obtained from the Kalman filter. Otherwise, there typically exists no closed form expression for the predictor. The conditional probabilities propagate according to Bayes rule which can be written as nonlinear partial difference equations, (e.g. Chapman-Kolmogorov) or the equivalent continuous time partial differential equations (Kushner-Stratonovich). Much effort has been spent over the years to find efficient (approximate) solutions to this *nonlinear filtering* problem. The 1990's saw a break-trough in these attempts. Markov Chain Monte Carlo (MCMC) methods, **[35], [23]** and more specifically Sequential Importance Sampling [33], developed into what is now generally known as *particle filters* (PF) and became a new efficient tool for nonlinear filtering. Intuitively it can be seen as approximate solutions of the underlying partial difference equations on a stochastically generated and carefully adapted grid in the x-space.

Another intuitive description is to see it as solving the equation (19a) for a (large) number of candidate solutions ("particles"), where solutions multiply stochastically, to mimic the noise term v. Each solution candidate is then matched to the observations y via (19b) to determine how relevant or important they are. This match is the basis of re-sampling the particles to keep the number constant (typically a few hundreds/thousands) and covering relevant areas of the state space.

A simplistic description of the basic process contains the following steps:

1) Use M "candidate solutions" (particles) to (19a), $x^i(t), i = 1, \dots, M$.
2) work, in principle, with approximations to the posterior probability density $q(t, x) = \pi_t(x(t)|Y^t)$ of the empirical distribution form
$$\hat{q}(t, x) = \sum_{i=1}^{M} \delta(x - x^i(t)) \tag{21a}$$
(δ denoting singleton distributions around the particles, implying that the distribution of the particles should mimic the posterior density)
3) After having observed $y(t)$, compute, for each particle, its (approximate) posterior probability $w^i(t) = P(x^i(t)|Y^t)$, by Bayes rule
$$w^i(t) = \frac{1}{\mathcal{N}} P(y(t)|x^i(t)) \tag{21b}$$
where $\mathcal{N}$ denotes normalization over all the M particles.
4) The posterior density of $x(t)$ given the observations is then approximated by the empirical distribution
$$\hat{q}(t, x) = \sum_{i=1}^{M} w^i(t)\delta(x - x^i(t)) \tag{21c}$$
5) Update the particles over time by drawing a sample of v in (19a).
6) Re-sample the particles at each time step according to the posterior weights w^i so that all the time equally weighted distributions (21a) are used for the next time step.

For a more comprehensive tutorial on particle methods we may refer to [24].

C. Application to Identification

Clearly, particle filters have opened up new avenues for nonlinear identification. The role of PFs both for off-line and on-line identification of nonlinear systems is discussed in [25] and [3], where expressions for the likelihood function and its gradient wrt θ are given based on PF calculations. A recent survey of nonlinear system identifiction using particle filtering is [42].

Another route to identification of nonlinear systems (19) is taken in [72]. Instead of computing the likelihood function and its gradient using particle filtering, which has a few technical problems, they employ the EM (Expectation-Maximization) method, [20], for estimation.

The EM algorithm is based on the iterations

$$\hat{\theta}_{k+1} = \arg\max_{\theta} Q(\theta, \theta_k) \tag{22a}$$

$$Q(\theta, \alpha) = E_\alpha[\log p_\theta(X^N, Y^N)|Y^N] \tag{22b}$$

where $p_\theta(X_N, Y_N)$ is the joint pdf of $X^N = \{x(1), \dots, x(N)\}$ and $Y^N = \{y(1), \dots, y(N)\}$ according to (19) and $E_\alpha(Z|Y^N)$ denotes conditional expectation of Z with respect to Y^N assuming Y^N is generated from (19) with $\theta = \alpha$.

The point now is that $Q(\theta, \theta_k)$ can readily be calculated from smoothed state estimates $E(x(t)|Y^N)$ assuming Y has been generated for the parameter value θ_k. In [72], it is shown how particle methods approximate the smoothed states sufficiently well, to yield good identification results. Recent study of smoothing with particle methods is given in [14], [22], and Section 5 in [46].

The example considered in [72] (and also in [25]) is

$$x(t+1) = \theta_1 x(t) + \theta_2 \frac{x(t)}{1 + x^2(t)} + \theta_3 \cos(1.2t) + \theta_4 v_t \tag{23a}$$

$$y(t) = \theta_5 x^2(t) + \theta_6 e(t) \tag{23b}$$

An interesting aspect of this example is that for $\theta_4 = 0$ (which is the case studied in [72]), the likelihood function can easily be calculated by just simulating (23a). Differentiating this equation w.r.t. θ_2 gives a difference equation that may be unstable. This means that the gradient of the likelihood function is very large at certain values, and that the likelihood function is highly multi-modal. Estimating the value of θ_4 (and finding it to be zero) and using the Q-function in (22a) instead of the likelihood function is therefore a good way out of problems with local minima.

The idea to use particle techniques to find smoothed state estimates, together with the EM methods has been applied to various nonlinear block-oriented models in [92] and [93].

D. A Variant: Minimum Distortion Filtering (MDF)

In [32] a deterministic choice of particles based on vector quantization is suggested instead of the randomly generated one in the PF. In short it can be described as the PF algorithm (21) with the following steps modified:

- Step 5 In the time update of the particles use a d-point approximation of v and thus expand, temporarily the M particles to $M \times d$ particles.
- Step 6 Instead of stochastically re-sampling the particles, use vector quantization (e.g. Lloyd's algorithm, [49]) to quantize back the $M \times d$ particles to M – taking the posterior weights into account.

This approach to nonlinear filtering has been further studied and tested in a number of papers, e.g. [31]. The potential of the MDF approach to filtering in system identification applications is particularly intriguing. See the promising examples in [16] and [17].

IV. System Identification Meets Sparsity

A. Preview

Sparse approximation and compressed sensing has been a vey active research area in the last few years, e.g. [15], [21]. Basically the problem can be described as follows: Given a matrix A, approximate it with a sparse matrix $\hat{A}$ (that has "many" elements equal to zero). So make the 2-norm $\|A - \hat{A}\|_2$ small while the ℓ_0-"norm" $\|\hat{A}\|_0$ is small (recall that the ℓ_0-"norm" $\|A\|_0$ means the number of non-zero elements of A). Various trade-offs between these competing minimizations are controlled by the criterion

$$\min \|A - \hat{A}\|^2 + \lambda \|\hat{A}\|_0 \tag{24}$$

($\|\cdot\|$ denotes the 2-norm: $\|\cdot\|_2$) depending on the size λ. Now, the problem (24) can be solved by postulating the number of nonzero elements of $\hat{A}$ (i.e. the number $\|\hat{A}\|_0$) and trying all the corresponding combinations of A-matrices. With n elements in A, this gives 2^n combinations to test. Clearly this is a forbidding task except for very small problems. A solution is to replace the ℓ_0 norm with a surrogate ℓ_1 norm (the sum of the absolute values of the entries)

$$\min \|A - \hat{A}\|^2 + \lambda \|\hat{A}\|_1 \tag{25}$$

This is now a convex criterion, which is easily minimized, and retains the feature that it favors solutions with many elements of $\hat{A}$ being exactly zero. This is, in short the basic idea about sparseness, and compressed sensing. The references [15], [21] contain technical results in what way the solution to (25) mimics the solution to (24).

B. Regressor Selection: LASSO

A long standing, and much discussed problem in estimation is the problem of regressor selection in linear regression:

$$Y = \Phi\theta + E \tag{26}$$

Each column in Φ corresponds to a regressor (and each row in θ to its corresponding parameter).

The regressor selection problem is to choose a suitable subselection of regressors that gives a good trade off between the model fit and the number of estimated parameters (cf. the Akaike criterion, [1])

$$\min_\theta \|Y - \Phi\theta\|^2 + \lambda\|\theta\|_0$$

This involves, as (24), a combinatorially increasing number of tests, which traditionally is handled by screening regressors in some *ad hoc* ordering (most relevant first, least relevant first, ...)

An alternative is to use the "ℓ_1-trick" and minimize the convex criterion

$$\min_\theta \|Y - \Phi\theta\|^2 + \lambda\|\theta\|_1 \tag{27}$$

which is known as *LASSO (Least Absolute Shrinkage Selection Operator)*, [77], and has been very successful in the past 15 years. An alternate term is *ℓ_1-regularization* since the criterion of fit has an additive penalty term on the size of the parameters, of the same type as the classical (Tikhonov or ℓ_2)-regularization with $+\lambda\|\theta\|_2^2$.

Groups of regressors: Group Lasso and Sum-Of-Norms (SON) Regularization: A variant of (26, 27) is the situation that the regressors can be grouped into several groups. We still want to use as few parameters as possible, but once an element in one group is used, it does not matter if the whole group is used. Let there be d groups, and use a ℓ_p-norm to measure the size of each group:

$$\Theta_k,\ k = 1, \ldots, d \tag{28a}$$

$$\Theta_k = \{\theta_{k,1}, \ldots \theta_{k,n_k}\} \tag{28b}$$

$$\|\Theta_k\|_p = \left(\sum_{j=1}^{n_k} \theta_{k,j}^p\right)^{1/p} \tag{28c}$$

$$n = \sum_{k=1}^{d} n_k \text{ is the total number of candidate parameters} \tag{28d}$$

Some parameter(s) in group k is used if and only if $\|\Theta_k\|_p > 0$. The number of groups used is thus

$$\|[\|\Theta_1\|_p, \ldots, \|\Theta_d\|_p]\|_0 \tag{29}$$

By relaxing this ℓ_0 norm to ℓ_1 we get the criterion

$$\min_\theta \|Y - \Phi\theta\|^2 + \lambda \sum_{k=1}^{d} \|\Theta_k\|_p \tag{30}$$

This is known as *Group Lasso*, [97], or Sum-Of-Norms regularization. We may note that if $p = 1$, this is the same as LASSO, so to obtain the group feature, it is essential to use a $p > 1$ in the group parameter norm.

As a system identification application, we may e.g. think of Y potentially modeling the response to d different inputs. With Group Lasso we can thus determine which inputs to really include in the modeling of Y.

C. Choosing the Regularization Parameter

The choice of the regularization parameter λ in (27) is usually done by way of cross-validation or generalized cross-validation; the regularization that performs best on fresh data is chosen. This requires solving (27) a number of times. However in [68] it is shown that AIC provides a way to directly obtain a reasonable value for λ. The idea is based on the observation that the full least squares estimate (let us denote it $\hat{\theta}_N^{LS}$) can model the true system. This suggests to find the estimate with smallest norm that has the same model fit as the full least-squares estimate. By model fit we here mean the fit on a fresh data set. According to [1] the expected least-squares cost on a new data set for the full least-squares estimate is given by

$$\left(1+\frac{2n}{N}\right) \mathrm{E}\left[\|Y-\Phi\hat{\theta}^{LS}\|^2\right] \tag{31}$$

where the expectation is over the new data. This suggests the following method

$$\begin{aligned} &\min_{\theta}\ \|\theta\|_1 \\ &\text{s.t. } \|Y-\Phi\theta\|^2 \leq \left(1+\frac{2n}{N}\right)\|Y-\Phi\hat{\theta}^{LS}\|^2 \end{aligned} \tag{32}$$

Ramifications and properties of this method can be found in [68]; see also [63] and [78] for related work.

D. Segmentation, LPV and Hybrid Models

Consider the problem to build a model of the following kind:

$$\hat{y}(t) = \theta^T(p(t))\varphi(t) + e(t) \tag{33}$$

where $\hat{y}$ is the predicted output, and $\varphi(t)$ is a vector of regression variables, known at time t. θ is a model parameter vector, that may depend on regime variable $p \in \mathbb{R}^p$, whose value at time t, $p(t)$, is known.

Typically $\theta(p)$ is a piecewise constant function of p:

$$\hat{y}(t) = \theta_k^T\varphi(t) \qquad \text{if } p(t) \in \mathcal{H}_k \tag{34a}$$

$$\mathcal{H}_k, k=1,\dots,d \quad \text{is a partition of } \mathbb{R}^p \tag{34b}$$

The class of models (33) is formally known as *linear parameter-varying, LPV* models, but depending on the regime variable it includes several other cases:

- If $\varphi(t)$ is formed from the recent past inputs and outputs, the model is of ARX type. If in addition φ contains a constant the model becomes affine.
- If $p = t$ = time, the model is a time-varying model. If θ is a piecewise constant function of t, we have a *segmented* (piece-wise constant) model.
- If $p(t) = \varphi(t)$ we have a piecewise linear or piecewise affine (PWA) model: in the partition $\mathcal{H}_k$ of the φ space the model is linear, but its coefficients change as the regression vector enters another region. This is a common case of a *hybrid* model. See e.g. [7].

There are a few requirements when such a model (34) is constructed:

1) Model estimation consists of
   a) finding d, the number of partitions in (34b).
   b) the d different parameter vectors θ_k.
   c) the expressions for the partitions $\mathcal{H}_k$
2) Use sufficiently large d to allow accurate description in different areas of the regime space.
3) Use sufficiently small d to avoid overfit and obtain a model that has reasonable complexity.

Suppose we have measured $y(t), u(t), t = \dots, N$ and want to find a model. To deal with Tasks 1a-1b, considering aspects 2 and 3 we could let loose one parameter vector at each sample and minimize

$$\sum_{t=1}^{N}\|y(t)-\theta_t^T\varphi(t)\|^2 + \lambda\sum_{s,t=1}^{N}K(p(t),p(s))\|\theta_s-\theta_t\|_0 \tag{35}$$

with respect to $\theta_t; t=1,\dots,N$.

The first term favors a good fit in accordance with requirement 2. The second term penalizes the number of different models in accordance with requirement 3. In (35) the kernel K is included to allow, for example, the possibility that different θ are not penalized if they correspond to "very different" regime variables.

Remark: The ℓ_0 norm in (35) should be interpreted in the group sense of (29), that is as 0-norm of the vector that is formed all the elements $\|\theta_t-\theta_s\|_p$. Actually, strictly speaking, this norm does not "count" the number of different θ_i: if there are d different models, such that model j is the same for k_j values of t ($N=\sum_{j=1}^d k_j$), the ℓ_0-norm takes the value $\sum_{i,j=1 i\neq j}^d k_i k_j$.

As before, the criterion (35) is computationally forbidding even for quite small N. We therefore relax the ℓ_0 norm to ℓ_1 norm as in (30) to obtain the convex Sum-Of-Norms criterion

$$\min_{\theta_t}\sum_{t=1}^{N}\|y(t)-\theta_t^T\varphi(t)\|^2 + \lambda\sum_{s,t=1}^{N}K(p(t),p(s))\|\theta_s-\theta_t\|_p \tag{36}$$

When (36) has been solved with a suitable λ that gives the desired trade-off between requirements 2 and 3, we have also solved 1a and 1b of requirement 1, and obtain

- d different parameter vectors $\Theta_k, k=1,\dots,d$, so each $\theta_t, t=1,\dots,N$ is equal to one of $\Theta_k, k=1\dots,d$.
- a clustering of the regime variable points:

$$p(t) \in P_k \text{ if } \theta_t \text{ is associated with model } \Theta_k \tag{37}$$

It now only remains to convert the points clusters P_k in the regime variable space to a partitioning $\mathcal{H}_k$ of this space:

$$P_k \subset \mathcal{H}_k \tag{38a}$$

$$\mathbb{R}^p = \cup_k\mathcal{H}_k \tag{38b}$$

This can be done with any one of many available clustering or pattern recognition algorithms, like e.g. the *Support Vector Machine* classifier, [81].

Example 5: Consider the multi-dimensional PWARX system (introduced in [6], see also [54], [7])

$$y_k = \begin{cases} -0.4y_{k-1} + & u_{k-1} + 1.5 + e_k, \text{ if} \\ & 4y_{k-1} - u_{k-1} + 10 < 0 \\ 0.5y_{k-1} - & u_{k-1} - 0.5 + e_k, \text{ if} \\ & 4y_{k-1} - u_{k-1} + 10 \geq 0 \text{ and} \\ & 5y_{k-1} + u_{k-1} - 6 < 0 \\ -0.3y_{k-1} + & 0.5u_{k-1} - 1.7 + e_k, \text{ if} \\ & 5y_{k-1} + u_{k-1} - 6 \geq 0. \end{cases} \tag{39}$$

Generate $\{u_k\}_{k=1}^{200}$ by sampling a uniform distribution $U(-4, 4)$ and let $e_k \sim U(-0.2, 0.2)$. Figure 5 shows the dataset $\{(y_k, u_k)\}_{k=1}^{200}$. The 200 data points thus correspond to 3 different models, such that y_k is a linear regression of $[u_{k-1}, y_{k-1}, 1]$ with 3 different coefficient vectors depending on the values of u_{k-1}, y_{k-1}. Figure 5 also shows how the algorithm (36) (for $\lambda = 0.01$) associates the regressors with 3 different regions (one for each parameter vector). The classification is 100% correct and we thus obtain the best possible estimates of the coefficients, and the best possible models.

As a comparison it is shown in Figure 6 how *Generalized Principal Component Analysis* (GPCA, [82]) performs on the same data. More details on this example are given in [57].

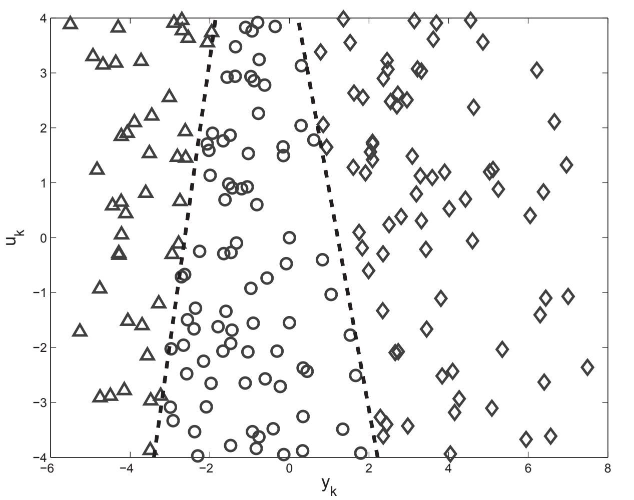

Fig. 5. Data used in Example 5 showed with △, ○ and ◇ symbols. The data marked with the same symbol got the same θ-estimate in (36). Dashed lines show the estimated partitions obtained by applying SVM. The true partitions coincide with the estimated ones.

Segmentation: Let us briefly comment on the segmentation case, i.e. when the regime-variable is a scalar and equal to time. If we only want to control the number of segments, i.e. the number of times the process parameters change, and do not insist on keeping the total number of different models small, it is natural in (36) only to penalize transitions, i.e. to let the kernel

$$K(p(t), p(s)) = K(t, s) = 0 \text{ unless } |t - s| = 1 \tag{40}$$

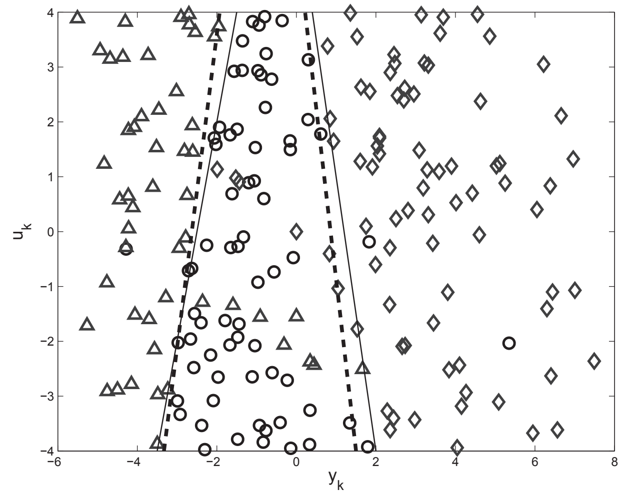

Fig. 6. The same as Figure 5 but using GPCA instead of the proposed method PWASON. 18 samples were misclassified. The shape of the partitions (dashed line) were estimated fairly well (true boundaries shown with solid thin line).

That means that the double sum in the regularization term collapses to a single sum:

$$\min_{\theta_t} \sum_{t=1}^{N} \|y(t) - \theta_t^T \varphi(t)\|^2 + \lambda \sum_{t=1}^{N-1} \|\theta_t - \theta_{t+1}\|_p \tag{41}$$

Example 6: Comparison between `segment` and (41)

Let us compare the method (41) with `segment` in the System Identification Toolbox [48]. Consider the system

$$\begin{aligned} y(t) + &a_1 y(t-1) + 0.7y(t-2) \\ &= u(t-1) + 0.5u(t-2) + e(t) \end{aligned} \tag{42}$$

with $u(t) \sim N(0, 1)$ and $e(t) \sim N(0, 9)$. At $t = 400$, a_1 changes from -1.5 to -1.3 and at $t = 1500$ a_1 returns to -1.5. Both `segment` and (41) are provided with the correct ARX structure and asked to estimate all ARX parameters (a_1, a_2, b_1, b_2). With the same design parameters as used to generate the data (the true equation error variance, jump probability, initial ARX parameters and covariance matrix of the parameter jumps) `segment` does not find any changes at all in the ARX parameters. Tuning the design variable R_2 in `segment` so it finds three segments gives the estimate of a_1 shown in Figure 7. It does not seem possible to find values of all the design variables in `segment` that give the correct jump instants.

Using (41) gives directly the correct change times, as seen in Figure 7.

See [59] for more on segmentation of ARX-models. A further example on segmentation of signals from nonlinear systems is given in [27].

E. State Smoothing with Abrupt Changes

The basic linear system with disturbances can be written

$$\begin{aligned} x(t+1) &= A_t x(t) + B_t u(t) + G_t v(t) \\ y(t) &= C_t x(t) + e(t). \end{aligned} \tag{43}$$

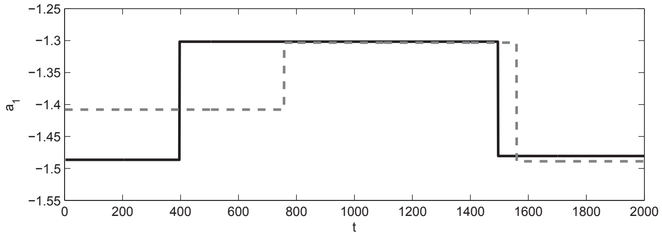

Fig. 7. Estimates of a_1 in the ARX-model used in Example 6 using (41) (solid) and `segment` (dashed).

Here, e is white measurement noise and v is a process disturbance. v is often modeled as Gaussian Noise which leads to the familiar Kalman filter state filtering and smoothing and the classical LQG control formulation

However in many applications, v is mostly zero, and strikes only occasionally:

$$v(t) = \delta(t)\eta(t)$$

where

$$\delta(t) = \begin{cases} 0 & \text{with probability } 1-\mu \\ 1 & \text{with probability } \mu \end{cases}$$
$$\eta(t) \in N(0, Q)$$

This is the case in many applications, like:

- Control: v are load disturbances acting as an unmeaured input. Pulse disturbances can be further shaped by the A-matrix to describe the actual load changes
- Tracking and path generation: v corresponds to unknown, sudden maneuvers to evade pursuers, or "knots" in the path curves
- Fault Detection and Isolation (FDI): v corresponds to additive system faults

The problem is to find the jump times t and/or the smoothed state estimates $\hat{x}_s(t|N)$. Over the years many different approaches have been suggested for this (nonlinear filtering) estimation problem. A sparse estimation approach is to use sum-of-norms regularization:

$$\min_{v(k),k=1,\dots,N-1} \sum_{t=1}^{N} \left\| \left(y(t) - Cx(t)\right) \right\|^2 + \lambda \sum_{t=1}^{N} \|v(t)\|_p$$
$$\text{s.t. } x(t+1) = Ax(t) + Bu(t) + Gv(t);\ x(1) = 0.$$

It performs quite well compared to traditional approaches, see, e.g., [55] and [56].

V. System Identification Meets Machine Learning

Machine learning has become a household word in the community dealing with inference, reasoning and actions based on data. The term is more broad and also more vague than the other encounters discussed here. The area has been growing and now typically incorporates general statistical tools for classification, pattern recognition, Gaussian Process Regression, kernel methods, sampling methods, unsupervised learning, etc. Some relevant books covering the topic include, [80], [10], [36], [66]. We shall in this section only describe some methods that have been used for System Identification Applications that stem from the Machine Learning Community.

A. Gaussian Process Regression for System Identification: General Ideas

A common problem in inference is to estimate functions from observations. For a Machine Learning perspective, see e.g. [66]. The problem is to estimate an n_f-dimensional function, say

$$f(\tau), \quad \tau \in \Omega \quad f(t) \in \mathbb{R}^{nf} \tag{44}$$

The domain Ω could be discrete or continuous time, or any, sufficiently regular, subset of $\mathbb{R}^d$. We observe

$$y(t), t = 1, \dots, N \tag{45}$$

that bear some information about the function f. The problem is to estimate f based on these observations.

One approach is to regard $f(\cdot)$ as a stochastic process, meaning that we can assign probability distributions to any vector $f(\tau_k), k = 1, \dots, n$ for any finite collection of points τ_k. From the observations y we can then compute the posterior distributions

$$p(f(\tau_k)|Y^N) \tag{46}$$

The calculation of (46) becomes easy if the prior distribution of f is a Gaussian Process, and the observations are linear function(al)s of f, measured in Gaussian noise:

$$y(t) = L(t, f) + e(t), \quad e(t) \in N(0, R) \tag{47}$$

That makes the posterior probabilities (46) also Gaussian, and the random vector $[y(t), f(\tau_k), k = 1, \dots, n]$ becomes a jointly Gaussian vector. Therefore the well known, and simple rules for conditional Gaussian probabilities can be applied when (46) is computed. This is the *Gaussian Processes Regression* approach. The book [66] contains many ramifications of this approach and Rasmussen has applied this technique to system identification, e.g. by estimating the state transition function for nonlinear systems. The spectacular web-site videos on learning to swing up and stabilize an inverted pendulum (`http://www.cs.washington.edu/homes/marc/learn_ctrl.html`) are based on this technique.

Hyper-parameters: For a successful application of Gaussian Process Regression it may be essential to have a good prior distribution for f, $p_0(f(\tau_k))$. It may be useful to equip this prior with some tuning parameters (*hyper-parameters*) α:

$$\text{Prior Distribution for } f: \qquad p_0(f(\tau_k), \alpha) \tag{48}$$

There are several techniques for tuning α. A basic approach is the so called *Empirical Bayes Method* which means that the distribution of the observations y is determined from (48) and (47). This distribution depends on α so this parameter can be estimated by the Maximum Likelihood method.

B. Identification of Linear systems

Pillonetto, de Nicolao and Chiuso have applied the Gaussian Process Regression perspective to the estimation of linear dynamical systems in several thought-provoking papers, e.g. [65], [64].

A linear system is completely characterized by its impulse response. Let us consider here, for simplicity, a discrete time system whose impulse response can be truncated after n values, without serious loss of accuracy. So we take the unknown function f to be estimated as the finite impulse response

$$f(\tau): \qquad g(k), k \in \Omega = \{1, \ldots, n\} \tag{49}$$

The response to any input u is

$$y(t) = \sum_{k=1}^{n} g(k)u(t-k) + v(t) \tag{50}$$

which is a particular case of (47). Introduce notations

$$y(t) = \varphi^T(t)\theta + e(t) \tag{51a}$$

$$\theta = \begin{bmatrix} g(1) & g(2) & \ldots & g(n) \end{bmatrix}^T \tag{51b}$$

$$\varphi(t) = \begin{bmatrix} u(t-1) & \ldots & u(t-n) \end{bmatrix}^T \tag{51c}$$

which can be written

$$Y = \Phi^T\theta + V \tag{51d}$$

with

$$Y = \begin{bmatrix} y(1) & y(2) & \ldots & y(N) \end{bmatrix}^T \tag{51e}$$

$$\Phi = \begin{bmatrix} \varphi^T(1) & \varphi^T(2) & \ldots & \varphi^T(N) \end{bmatrix}^T \tag{51f}$$

$$V = \begin{bmatrix} v(1) & v(2) & \ldots & v(N) \end{bmatrix}^T \quad E\, VV^T = \sigma^2 I \tag{51g}$$

If the prior distribution of θ (eqs (48), (49)) is

$$\theta \in N(0, P_0(\alpha)) \tag{52}$$

then, the posterior mean, given Y is well known to be (see e.g. [19])

$$\hat{\theta}^{\text{apost}} = (\Phi\Phi^T + \sigma^2 P_0(\alpha)^{-1})^{-1}\Phi Y \tag{53}$$

This "Gaussian Process" estimate we also recognize as the regularized least squares estimate

$$\theta^{\text{apost}} = \arg\min_{\theta} \|Y - \Phi\theta\|^2 + \theta^T\sigma^2 P_0(\alpha)^{-1}\theta \tag{54}$$

So, the Gaussian Process Regression estimate for linear systems is not a spectacular or a truly innovative result. The exciting aspect of (53) is that for carefully chosen hyperparameters α it may performs many conventional linear system identification techniques. See Figure 8, and the papers [65], [19] for more details and discussions.

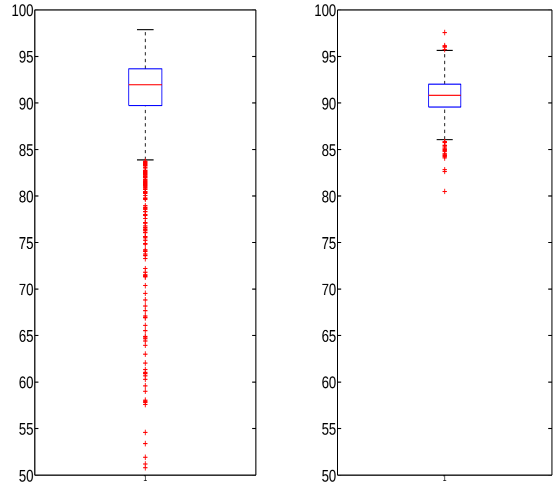

Fig. 8. Box-plots for the 2500 fits for a randomly generated data set with random systems of high orders. The fit shows how well the model can reproduce the true system. 100% fit means a perfect model. The left figure shows a straightforward application of the OE (output error) method and the right figure shows the regularized FIR model (54).

C. Manifold Learning and Unsupervised Learning

The basic estimation and identification model can often be written as a regression

$$y = f(\varphi) \tag{55}$$

where φ is a vector of observed variables, the *regressors*, and y is the object of interest, the *output*. The observations of interest can be a collection of pairs

$$y(t) = \varphi(t) + e(t), \quad t = 1, \ldots, N_e \tag{56}$$

were e accounts for possible errors in the observed outputs, and/or a collection of relevant regression vectors:

$$Z_u = \{\varphi(t), t = N_e + 1, \ldots, N_e + N_u\} \tag{57}$$

This are often referred to as the *unlabeled regressors*. The objective is to learn the mapping f, so that for any relevant regressor φ^* we can associate a corresponding value y^*. Clearly the difficulty of this task depends (among other things) on the complexity and size of the region (space) where the regressors take their values. We denote this region by $\mathcal{D}$:

$$\varphi \in \mathcal{D} \tag{58}$$

Manifold learning and *semi-supervised learning* are two central concepts in machine learning, [70], [96], [18]. With a very brief and simplistic definition, manifold learning can be described as the task to infer $\mathcal{D}$ from (57), and semi-supervised learning concerns using (57) (together with (56)) to obtain a better estimate of f.

In the machine learning literature, non-parametric approaches are common. This means that the regression function is allowed to fit very freely to data, at the same time as some regularization is used to curb the freedom. [This is actually also the idea behind the simple FIR model (54).] Let us suppose that we would liketo g estimate $f(\varphi(t))$ for any $\varphi(t)$ in Z_u, (57), which may be extended for this purpose. With extreme amount of freedom we associate a separate estimate for each regressor: Let $\hat{f}_t$ correspond to the output for regressor value $\varphi(t)$. That gives a criterion

$$V(\hat{f}) = \sum_{t=1}^{N_e} (y(t) - \hat{f}_t)^2 \tag{59}$$

to be oprimized over $\hat{f}_t$, This is clearly too flexible a fit. We must weigh that against a wish that the output depends smoothly on the regressors in the relevant regions. Using a kernel, we can express this as

$$\hat{f}_t = \sum_{i=1}^{N_e+N_u} K_{ti} \hat{f}_i, \quad t = 1 \dots N_e + N_u \tag{60}$$

where K_{ti} is a kernel giving a measure of distance between $\varphi(t)$ and $\varphi(i)$, relevant to the assumed region. So the sought estimates $\hat{f}_i$ should be such that they are smooth over the region. At the same time, for regressors with measured labels, the estimates should be close to those, meaning that (59) should be small. The two requirements (60) and (59) can be combined into a criterion

$$\lambda \sum_{i=1}^{N_e+N_u} \left(\hat{f}_i - \sum_{j=1}^{N_e+N_u} K_{ij} \hat{f}_j \right)^2 + (1-\lambda) \sum_{t=1}^{N_e} (y(t) - \hat{f}_t)^2 \tag{61}$$

to be minimized with respect to $\hat{f}_t$, $t = 1, \dots, N_e + N_u$. The scalar λ decides how trustworthy our labels are and is seen as a design parameter.

The criterion (61) can be given a Bayesian interpretation as a way to estimate $\hat{f}$ in (59) with a “smoothness prior” (60), with λ reflecting the confidence in the prior.

Introducing the notation

$$\begin{aligned} J &\triangleq [I_{N_e \times N_e} \; 0_{N_e \times N_u}], \\ \vec{y} &\triangleq [y(1) \, y(2) \dots y(N_e)]^T, \\ \vec{\hat{f}} &\triangleq [\hat{f}_1 \, \hat{f}_2 \dots \hat{f}_{N_e} \, \hat{f}_{N_e+1} \, \dots \hat{f}_{N_e+N_u}]^T, \end{aligned}$$

$$K \triangleq \begin{bmatrix} K_{11} & K_{12} & \dots & K_{1,N_e+N_u} \\ K_{21} & K_{22} & & K_{2,N_e+N_u} \\ \vdots & & \ddots & \vdots \\ K_{N_e+N_u 1} & K_{N_e+N_u,2} & \dots & K_{N_e+N_u,N_e+N_u} \end{bmatrix},$$

(61) can be written as

$$\lambda(\vec{\hat{f}} - K\vec{\hat{f}})^T(\vec{\hat{f}} - K\vec{\hat{f}}) - (1-\lambda)(\vec{y} - J\vec{\hat{f}})^T(\vec{y} - J\vec{\hat{f}}) \tag{62}$$

which expands into

$$\begin{aligned} &\vec{\hat{f}}^T \left(\lambda(I - K - K^T + K^T K) - (1-\lambda) J^T J \right) \vec{\hat{f}} \\ &\quad + 2(1-\lambda) \vec{\hat{f}}^T J^T \vec{y} + (1-\lambda) \vec{y}^T \vec{y}. \end{aligned} \tag{63}$$

Setting the derivative with respect to $\vec{\hat{f}}$ to zero and solving gives the linear kernel smoother

$$\begin{aligned} \vec{\hat{f}} =& (1-\lambda) \left(\lambda(I - K - K^T + K^T K) \right. \\ & \left. - (1-\lambda) J^T J \right)^{-1} J^T \vec{y}. \end{aligned} \tag{64}$$

This regression procedure uses all regressors, both unlabeled and labeled, and is hence a semi-supervised regression algorithm. We call the kernel smoother *Weight Determination by Manifold Regularization* (WDMR, [61]). In this case the unlabeled regressors are used to get a better knowledge for what parts of the regressor space that the function f varies smoothly in.

The papers [60] and [58] contain several examples of how WDMR behaves on estimation and identification problems.

D. *Reinforcement Learning*

Inspiration from the way living organisms improve their actions based on the outcomes of previous actions has lead to the development of a branch of machine learning called reinforcement learning (RL) [53]. As the learning objective is formulated in terms of the minimization of a cost function, the theoretical foundations of this area are rooted in optimal control. As compared to standard optimal control there are two main differences:

- Whereas the Bellman optimality equations provide a solution going backwards in time, RL algorithms learn in real time meaning that the solution has to be found going forwards in time.
- Parts of the, or the entire, system equations are unknown in RL; necessary system information has to be acquired by taking actions and observing the system response.

On selected problems RL has shown impressive performance, e.g. the RL backgammon algorithm in [76]. The inverted pendulum stabilization referred to in Section V-A is another example showing the potential. We refer to the excellent tutorial [45] and to [74], [8] for further details on RL. A rich set of algorithms have been developed and go under various names, e.g. approximate dynamic programming and neuro-dynamic programming. Here we will discuss some connections between RL and optimal experiment design for system identification. We begin with reviewing the basics of RL; restricting attention to deterministic linear quadratic (LQ) control. The algorithm we will study is called heuristic dynamic programming (HDP) [91].

Consider minimizing the quadratic cost

$$J(x_0) := \sum_{t=0}^{\infty} \left(x^T(t) Q x(t) + u^2(t) \right), \; Q = Q^T > 0, \tag{65}$$

for the single-input single-output linear time-invariant system

$$x(t+1) = Ax(t) + Bu(t), \quad x(0) = x_o \in \mathbb{R}^n \tag{66}$$

For a given feedback $u(t) = -Kx(t)$, we have

$$J(x_o) = x_o^T P(A, B, K) x_o \tag{67}$$

where $P(A,B,K)$ is the solution to the discrete Lyapunov equation

$$P(A,B,K) = \\ Q + K^T K + (A-BK)^T P(A,B,K)(A-BK) \quad (68)$$

The well known optimal solution to

$$J_o(x_0) := \min_{u(0),u(1),\dots} J(x_0) \quad (69)$$

is given by $u = -\bar{K}x$ where

$$\bar{K} := (1 + B^T \bar{P} B)^{-1} B^T \bar{P} A \quad (70)$$

where $\bar{P} = P(A,B,\bar{K})$ is obtained by solving (68) (which with (70) inserted now becomes a Riccati equation).

The basic HDP algorithm for the LQ problem is given by[1]

i) Take $P_0 = 0$ and set $j = 0$.
ii) Take $K_j = (1 + B^T P_j B)^{-1} B^T P_j A$
iii) Take $P_{j+1} = Q + K_j^T K_j + (A - BK_j)^T P_j (A - BK_j)$
iv) Set $j \to j+1$ and go to Step ii).

Notice that unlike the Ricatti equation for the LQ problem, these equations evolve forward in time. It has been shown that these iterations converge to the optimal control policy, i.e. $P_j \to \bar{P}$ and $K_j \to \bar{K}$ [2].

The iterations above require knowledge of the system matrices A and B. A key ingredient in RL is to by-pass this requirement using measurements. The idea is to replace the Updates ii) and iii) in Iteration j by updates based on experimental data collected with the current control gain K_j. This is closely related to direct adaptive control where the system is modeled in terms of its optimal controller – in the LQ case $\bar{P}$ and $\bar{K}$.

There are various strategies for the exploration that takes place during the data collection in each iteration. Here we observe interesting links to applications-oriented optimal experiment design (AOED) in system identification. AOED concerns how to design the system identification experiment such that, given experimental constraints, the best possible performance is obtained in the application for which the identified model is used [38]. We will briefly discuss some notions that have emerged during the study of this problem which seem to relate to RL.

The introduction of least-costly identification [11] lead to the realization that not only does optimal identification experiments enhance the visibility of system properties of importance in the measurements, but also, in order to reduce the experimental cost, they avoid exciting properties that are irrelevant to the application [37], [38]. This means that when such experimental conditions are used the system identification problem is simplified as irrelevant properties then do not have to be modeled. We will illustrate the concept with an example.

Example 7: Consider the system (66). Suppose now that both system matrices A and B are unknown but that noise free observations of $x(t)$ can be obtained and consider the problem of identifying the minimum LQ cost $J_o(x_o)$ from such observations. Let us also assume that the optimal gain $\bar{K}$ is available to the user. This assumption is of course not realistic but will be helpful for the issue we are trying to highlight in this example. In view of that (65) measures the performance of the system it is natural to measure the cost of the identification experiment by

$$J_{id} := \sum_{t=0}^{\infty} \left(x_{id}^T(t) Q x_{id}(t) + u_{id}^2(t) \right), \quad (71)$$

where $\{u_{id}(t)\}_{t=0}^{\infty}$ denotes the input sequence used during the identification experiment and where $\{x_{id}(t)\}_{t=0}^{\infty}$ denotes the resulting state sequence.

We now pose the following optimal experiment design problem: What is the minimum identification cost, as measured by (71), required to get a perfect estimate of $J_o(x_o)$ when starting in state x_o? Well, since, $J_o(x_o)$ is the minimum achievable cost we have $J_{id} \geq J_o(x_o)$ but then we immediately realize that if we use the optimal control (achieving $J_o(x_o)$), then we can get a perfect estimate of $J_o(x_o)$ simply by computing $\sum_{t=0}^{\infty} x_{id}^T(t) Q x_{id}(t) + u_{id}^2(t)$ using the observed (noise-free) state-sequence $x_{id}(t)$ and the corresponding applied controls $u_{id}(t)$.

The optimal solution has two features: Firstly, the desired quantity, in this case $J_o(x_o)$, is clearly observable from the data. Secondly, all properties of the system irrelevant to the objective are hidden by the experiment. Notice that the system matrices are not identifiable with this experiment as the system evolves according to

$$x(t+1) = A_{cl} x(t), \quad x_0 = x_o \quad (72)$$

where $A_{cl} = A - B\bar{K}$, thus only A_{cl} can be identified. However, from the expression for $P(A,B,K)$, (68), we see that A_{cl} is exactly the system property required to estimate $J_o(x_o)$. ■

RL/AOED Link 1: Applications specific models: The second feature in Example 7 is due to that we are minimizing an identification cost (71) that is closely related to the application. The implication of this feature is that the identification problem is simplified as we only have to model the features that are relevant to our objective (that of estimating $J_o(x_o)$ in this example). Here we do not even have to estimate A and B, we just have to sum up the observations according to J_{id}!

Now let us return to the RL algorithm i)–iv) above where P_j and K_j can be seen as model parameters. Notice that the Riccati map $(A,B) \to (\bar{P},\bar{K})$ defined by (69) and (70) is not bijective. That means that in RL only system properties relevant for the (optimal control) application are modelled. This thus corresponds very closely to the outcome of the optimal experiment design problem in Example 7.

The use of applications specific models is potentially interesting for applications oriented modeling of complex systems. The quality of the model can then be governed by the performance demands of the application. Also the problem of overmodelling is mitigated when optimal experiments are performed. See [38] for further discussion.

[1] See the appendix for details.

RL/AOED Link 2: Exploration strategies: Another interesting link between RL and AOED lies in system exploration. AOED is a systematic way to precisely reveal the necessary system information for the application at hand. Hence, it may be of interest to study whether ideas from AOED, such as computational algorithms, e.g. [41], but also theoretical considerations such as the assessment of the cost of exploration [69], can fit into the RL framework. Conversely exploration strategies developed within the RL framework may very well have potential in AOED.

RL/AOED Link 3: Adaptation strategies: It has been noted for certain applications that it is good to perform identification experiments under the desired operating conditions. Example 7 is one such example; for some control applications see [30], [38]. Our next example, which extends Example 7, reinforces this notion.

Example 8: Consider again the system (66) and the problem of estimating (69) based on measurements of the system. However, assume now that the measurements are noisy, i.e. we can observe

$$y(t) = x(t) + e(t) \tag{73}$$

where $\{e(t)\}$ is zero mean Gaussian white noise with covariance matrix $\lambda\, I$, for some $\lambda > 0$ (I is the $n \times n$ identity matrix). In this case we can obviously not recover $J_o(x_o)$ exactly from the measurements $\{y(t)\}$. We will discuss two different methods to estimate $J_o(x_o)$ in this case. We will assume that the optimal feedback $\bar{K}$ is known. In the first approach we will use optimal open loop experiment design and estimate the system matrices A and B explicitly. In the design we minimize the expected value of the squared error $(x_o^T P(\hat{A}, \hat{B}, \bar{K}) x_o - J_o(x_o))^2$ subject to that the average of the experimental energy as measured by (71) is bounded by $J_o(x_o)$. We refer to the appendix for details.

The second way of estimating $J_o(x_o)$ consists of three steps. First generate data using the optimal state feedback controller

$$u(t) = -\bar{K}x(t)$$

This means that the system evolves according to (72) and (73). After this experiment, use the observations $\{y(t)\}$ to estimate the closed loop system matrix from the equations (72). Here, the prediction error estimate is given by

$$\hat{A}_{cl} = \arg\min_{\bar{A}} \sum_{t=1}^{N} (y(t) - \bar{A}^t x_o)^2 \tag{74}$$

Finally, from $\hat{A}_{cl}$ an estimate of $\bar{P}$ is obtained by solving (68) with $A - B\bar{K} = \hat{A}_{cl}$. Then using (67), with $P(A, B, \bar{K})$ substituted for the estimate, gives an estimate of $J_o(x_o)$. Notice that since the optimal feedback $\bar{K}$ is used but without external excitation the same identification cost as in the optimal experiment design problem (83) is incurred.

Running 1000 Monte Carlo simulations with noise variance $\lambda = 0.01$ when

$$A = \begin{bmatrix} 0 & 0 \\ 1 & 0.8 \end{bmatrix},\ B = \begin{bmatrix} 1 \\ 0 \end{bmatrix},\ x_o = \begin{bmatrix} 1 \\ 0 \end{bmatrix},\ Q = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} \tag{75}$$

gives that the average quality of this estimate is within a few percent of the average quality obtained when A and B are estimated using the optimal open loop experiment. We find this quite intriguing as in the second approach no external excitation is used in the experiment; it is only the transient, due to the non-zero initial state x_o, that is observed in noise. We conclude that, as in Example 7, the experiment generated by the optimal controller has very nice properties from an identification perspective. ■

As the optimal operating conditions in general depend on the true system, it is typically impossible to perform these types of experiments. To overcome this problem, adaptive methods have been proposed, e.g. [29]. This is also one of the tenets of so called iterative identification and control, e.g. [99]. We now observe that RL is designed to (eventually) achieve optimal operating conditions. This suggests that RL algorithms have an interesting potential in AOED.

Rapprochement and outlook: Both RL and identification for control can be seen as substitutes for the computationally infeasible dual control. The areas have focused on different aspects of this difficult problem and an amalgamation of the ideas from these fields could provide a significant push forward to the problem of autonomous learning a control objective for complex systems. Above, we have pointed to a few possible directions into unchartered territory.

VI. Conclusion: The Role of System Identification

It is clear to everyone in science and engineering that *mathematical models* are playing increasingly important roles. Today, model-based design and optimization is the dominant engineering paradigm to systematic design and maintenance of engineering systems. In control, the aerospace industry has a long tradition of model based design and in the process industry Model Predictive Control has become the dominant method to optimize production on intermediate levels. Also, driven by the "grand" challenges society are facing, e.g. energy and environmental considerations, new model based control applications are emerging *en masse*: Automotive systems, power grids, and medical systems are but a few examples of areas where funding agencies and industry worldwide are devoting massive investments at the moment. These new applications represent highly complex systems with high demands on autonomy, and adaptation. The models used internally in these systems thus also need to be maintained and updated autonomously calling for data driven models.

In the process industry it has been observed that that obtaining the model is the single most time consuming task in the application of model-based control and that three quarters of the total costs associated with advanced control projects can be attributed to modeling. This hints that *modeling risks becoming a serious bottleneck in future engineering systems.*

It is therefore vital that the area of System Identification is able to meet the challenges from model-based engineering to provide the necessary tools. Certainly, these challenges will be strong drivers for research in the field in the years to come. We have in this contribution pointed to how System Identification in recent years has encountered four other research areas and been able to amalgamate essential features of them to produce sharpened tools for model estimation.

ACKNOWLEDGMENTS: This work was supported by the European Research Council under the advanced grant LEARN, contract 267381. The authors also thank Graham Goodwin, Thomas Schön, Cristian Rojas and Carlo Fischione for helpful comments on the manuscript.:

REFERENCES

[1] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, AC-19:716–723, 1974.

[2] A. Al-Tamimi, F.L. Lewis, and M. Abu-Khalaf. Discrete-time nonlinear HJB solution using approximate dynamic programming: Convergence proof. *IEEE Trans. Man, and Cyb. – Part B: Cyb.*, 38(4):943–949, Aug 2008.

[3] C. Andrieu, A. Doucet, S. S. Singh, and V. B. Tadić. Particle methods for change detection, system identification and control. *Proceeding of IEEE*, 92(3):423–438, 2004.

[4] J. Baillieul and P. J. Antsaklis. Control and communication challenges in networked real-time systems. *Proc. of the IEEE*, 95(1):9–28, Jan. 2007.

[5] S. Barbarossa and G. Scutari. Decentralized maximum-likelihood estimation for sensor networks composed of nonlinearly coupled dynamical systems. *IEEE Trans. Signal Proc.*, 55(7):3456–3470, Jul. 2007.

[6] A. Bemporad, A. Garulli, S. Paoletti, and A. Vicino. A greedy approach to identification of piecewise affine models. In *Proceedings of the 6th international conference on Hybrid systems (HSCC'03)*, pages 97–112, Prague, Czech Republic, 2003. Springer-Verlag.

[7] A. Bemporad, A. Garulli, S. Paoletti, and A. Vicino. A bounded-error approach to piecewise affine system identification. *IEEE Transactions on Automatic Control*, 50(10):1567–1580, October 2005.

[8] D. P. Bertsekas and J. Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, 1996.

[9] D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Prentice-Hall, New York, 1989.

[10] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[11] X. Bombois, G. Scorletti, M. Gevers, P. M. J. Van den Hof, and R. Hildebrand. Least costly identification experiment for control. *Automatica*, 42(10):1651–1662, 2006.

[12] V. Borkar and P. P. Varaiya. Asymptotic agreement in distributed estimation. *IEEE Trans. Aut. Control*, 27(3):650–655, Jun. 1982.

[13] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–124, 2011.

[14] M. Briers, A. Doucet, and S. Maskell. Smoothing algorithms for state-space models. *Ann. Inst. Stat. Math.*, 52:61–89, 2010.

[15] E. J. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52:489–509, February 2006.

[16] M. G. Cea and G. C Goodwin. A comparison of particle filtering and minimum distortion filtering algorithms for nonlinear filtering. In *The IEEE Conference on Decision and Control, CDC*, Orlando, FL, Dec 2011.

[17] M. G. Cea and G. C. Goodwin. A new paradigm for state estimation in nonlinear systems via minimum distortion filering. In *Proc. IFAC World Congress*, Milan, Italy, Aug 2011.

[18] O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006.

[19] Tianshi Chen, Henrik Ohlsson, and Lennart Ljung. On the estimation of transfer functions, regularizations and gaussian processes – revisited. In *Proc. IFAC World Congress*, volume NA, page NA, Milan, Italy, August 2011.

[20] A. P. Demster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Royal Stat. Soc. B*, 39(1):1–39, 1977.

[21] D. L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, April 2006.

[22] R. Douc, E. Moulines, and J. Olsson. Optimality of the auxiliary particle filter. *Probability and Mathematical Statistics*, 29:1–28, 2009.

[23] A. Doucet, J. F. G. de Freitas, and N. J. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer, New York, 2001.

[24] A. Doucet and A. Johansen. A tutorial in particle filtering and smoothing: Fifteen years later. In D. Crisan and B. Rozovsky, editors, *The Oxford Handbook od Nonlinear Filtering*. Oxford University Press, 2011.

[25] A. Doucet and V. B. Tadić. Parameter estimation in general state-space models using particle methods. *Ann. Inst. Statist. Math.*, 55(2):409–422, 2003.

[26] F. Eng and F. Gustafsson. Identification with stochastic sampling time jitter. *Automatica*, 44(3):637–646, 2008.

[27] T. Falck, H. Ohlsson, L. Ljung, J. A.K. Suykens, and B. De Moor. Segmentation of times series from nonlinear dynamical systems. In *Proceedings of the 18th IFAC World Congress*, Milano, Italy, 2011.

[28] K. F. Gauss. *Theoria Motus Corporum Celestium, English Translation: Theory of the Motion of Heavenly Bodies*. Dover (1963), New York, 1809.

[29] L. Gerencsér, H. Hjalmarsson, and J. Mårtensson. Identification of ARX systems with non-stationary inputs - asymptotic analysis with application to adaptive input design. *Automatica*, 45(3):623–633, March 2009.

[30] M. Gevers and L. Ljung. Optimal experiment designs with respect to the intended model application. *Automatica*, 22(5):543–554, 1986.

[31] G. C. Goodwin and M. G. Cea. Continuous and discrete nonlinear filtering: Parts 1&2. *Automatica*, 2011. Submitted.

[32] G. C. Goodwin, A. Feuer, and C. Müller. Sequential bayesian filtering via minmum distortion quantization. In X. Hu, U. Jönsson, B. Wahlberg, and B.Ghosh, editors, *Three Decades of Progress in Systems and Control*. Springer Verlag, Berlin, 2010. Jan.

[33] N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear non-gaussian Bayesian state estimation. *IEE Proceedings F*, 140:107–113, 1993.

[34] T. S. Han and S. Amari. Statistical inference under multiterminal data compression. *IEEE Trans. Inf. Theory*, 44(6):2300–2324, oct 1998.

[35] J. E. Handschin and D. Q. Mayne. Monte Carlo techniques to estimate the conditional expectation in multi-stage non-linear filtering. *Int. J. of Control*, 9(5):547–559, 1969.

[36] T. Hastie, R Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

[37] H. Hjalmarsson. From experiment design to closed loop control. *Automatica*, 41(3):393–438, March 2005.

[38] H. Hjalmarsson. System identification of complex and structured systems. *European Journal of Control*, 15(4):275–310, 2009. Plenary address. European Control Conference.

[39] K. Jacobsson and H. Hjalmarsson. Closed loop aspects of fluid flow model identification in congestion control. In *14th IFAC Symposium on System Identification*, Newcastle, Australia, March 2006.

[40] Krister Jacobsson, Lachlan L. H. Andrew, A. Kevin Tang, Steven H. Low, and H. Hjalmarsson. An improved link model for window flow control and its application to FAST TCP. *IEEE Transactions on Automatic Control*, 54(3):551–564, 2009.

[41] H. Jansson and H. Hjalmarsson. Input design via LMIs admitting frequency-wise model specifications in confidence regions. *IEEE Transactions on Automatic Control*, 50(10):1534–1549, 2005.

[42] N. Kantas, A. Doucet, S. S. Singh, and J. M. Maciejowski. An overview of sequential Monte Carlo methods for parameter estimation in general state-space models. In *Proc. 15th IFAC Symposium on System Identification*, Saint-Malo, France, July 2009.

[43] F. Kelly, A. Maulloo, and D. Tan. Rate control in communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research Society*, 49:237–252, 1998.

[44] E. L. Lehmann and G. Casella. *Theory of Point Estimation*. John Wiley & Sons, New York, second edition, 1998.

[45] F. L. Lewis and D. Vrabie. Reinforcement learning and adaptive dynamic programming for feedback control. *IEEE Circ. and Syst. Mag.*, 9(3):32–50, . 2009.

[46] F. Lindsten. Rao-blackwellised particle methods for inference and identification. Technical Report Licentiate thesis 1480, Dept. of Electrical Engineering, Linköping University, Sweden, June 2011.

[47] L. Ljung. *System Identification: Theory for the User*. Prentice-Hall, Englewood Cliffs, NJ, 2nd edition edition, 1999.

[48] L. Ljung. *The System Identification Toolbox: The Manual*. The MathWorks Inc. 1st edition 1986, 7th edition 2007, Natick, MA, USA, 2007.

[49] S. Lloyd. Least squares quantization in PCM. *IEEE Trans. Information Theory*, IT-28:127–135, March 1982.

[50] S. H. Low, F. Paganini, and J. C. Doyle. Internet congestion control. *Control Systems Magazine*, 22(1):28–43, February 2002.

[51] S. H. Low and R. Srikant. A mathematical framework for designing a low-loss, low-delay Internet. *Networks and Spatial Economics*, 4:75–101, 2004.

[52] Z.-Q. Luo, M. Gatspar, J. Liu, and A. Swami, editors. *IEEE Signal Processing Magazine: Special Issue on Distributed Signal Processing in Sensor Networks*. IEEE, July 2006.

[53] J.M. Mendel and R.W. MacLaren. Reinforcement learning control and pattern recognition systems. In J.M. Mendel and K.S. Fu, editors, *Adaptive, Learning and Pattern Recognition Systems: Theory and Applications*, pages 287–318. Academic Press, New York, 1970.

[54] H. Nakada, K. Takaba, and T. Katayama. Identification of piecewise affine systems based on statistical clustering technique. *Automatica*, 41(5):905–913, 2005.

[55] H. Ohlsson, F. Gustafsson, L. Ljung, and S. Boyd. State smoothing by sum-of-norms regularization. In *Proceedings of the 49th IEEE Conference on Decision and Control*, Atlanta, USA, December 2010.

[56] H. Ohlsson, F. Gustafsson, L. Ljung, and S. Boyd. Trajectory generation using sum-of-norms regularization. In *Proceedings of the 49th IEEE Conference on Decision and Control*, Atlanta, USA, December 2010.

[57] H. Ohlsson and L. Ljung. Piecewise affine system identification using sum-of-norms regularization. In *Proceedings of the 18th IFAC World Congress*, Milano, Italy, 2011.

[58] H. Ohlsson and L. Ljung. Weight determination by manifold regularization. In A. Rantzer and R. Johansson, editors, *Distributed Decision Making and Control*. Springer Verlag, 2011.

[59] H. Ohlsson, L. Ljung, and S. Boyd. Segmentation of ARX-models using sum-of-norms regularization. *Automatica*, 46(6):1107–1111, 2010.

[60] Henrik Ohlsson and Lennart Ljung. Semi-supervised regression and system identification. In B. Wahlberg X. Hu, U. Jonsson and B.Ghosh, editors, *Three Decades of Progress in Systems and Control*. Springer Verlag, January 2010.

[61] Henrik Ohlsson, Jacob Roll, and Lennart Ljung. Manifold-constrained regressors in system identification. In *Proc. 47st IEEE Conference on Decision and Control*, pages 1364–1369, December 2008.

[62] H. Papadopoulos, G. Wornell, and A. Oppenheim. Sequential signal encoding from noisy measurements using quantizers with dynamic bias control. *IEEE Trans. Inf. Theory*, 47(3):978–1002, Mar. 2001.

[63] K. Pelckmans, J. A. K. Suykens, and B. De Moor. Additive regularization trade-off: Fusion of training and validation levels in kernel methods. *Machine Learning*, 62(3):217–252, mar 2006.

[64] G. Pillonetto, A. Chiuso, and G. De Nicolao. Prediction error identification of linear systems: a nonparametric Gaussian regression approach. *Automatica*, 47(2):291–305, 2011.

[65] G. Pillonetto and G. De Nicolao. A new kernel-based approach for linear system identification. *Automatica*, 46(1):81–93, January 2010.

[66] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, December 2005.

[67] A. Ribeiro and G. B. Giannakis. Bandwidth-constrained distributed estimation for wireless sensor networks-part i: Gaussian case. *IEEE Trans. Signal Proc.*, 54(3):1131–1143, Mar. 2006.

[68] C. R. Rojas and H. Hjalmarsson. Sparse estimation based on a validation criterion. In *Proceedings 49th IEEE Conference on Decision and Control*, Orlando, FA, USA, 2011. Submitted.

[69] C. R. Rojas, M. Barenthin Syberg, J.S. Welsh, and H. Hjalmarsson. The cost of complexity in system identification: Frequency function estimation of Finite Impulse Response systems. *IEEE Transactions on Automatic Control*, 55(10):2298–2309, 2010.

[70] Sam T. Roweis and Lawrence K. Saul. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science*, 290(5500):2323–2326, 2000.

[71] S. Roy and R. A. Iltis. Decentralized linear estimation in correlated measurement noise. *IEEE Trans. Aerosp. and Electr. Syst.*, 27(6):939–941, Nov. 1991.

[72] Thomas B. Schön, Adrian Wills, and B. Ninness. System identification of nonlinear state-space models. *Automatica*, 47(1):39–49, January 2011.

[73] R. Srikant. *The Mathematics of Internet Congestion Control*. Birkhauser, 2004.

[74] R.S. Sutton and A.G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, 1998.

[75] A. Tang, L.L.H. Andrew, K. Jacobsson, K.H. Johansson, H. Hjalmarsson, and S.H. Low. Queue dynamics with window flow control. *IEEE/ACM Transaction on Networking*, 18(5):1422–1435, oct 2010.

[76] G. Tesauro. Temporal difference learning and TD-gammon. *Communications of the ACM*, 38(3):58–68, mar 1995.

[77] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of Royal Statistical Society B (Methodological)*, 58(1):267–288, 1996.

[78] R. Tóth, B. S. Sanandaji, K. Poolla, and T. L. Vincent. Compressive system identification in the linear time-invariant framework. In *Proceedings 40th IEEE Conference on Decision and Control*, Orlando, Florida, USA, December 2011. Submitted.

[79] J. N. Tsitsiklis, D. P. Bertsekas, and M. Athans. Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE Trans. Aut. Control*, 31(9):803–812, Sep. 1986.

[80] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, 1995.

[81] V. N. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.

[82] R. Vidal, S. Soatto, Y. Ma, and S. Sastry. An algebraic geometric approach to the identification of a class of linear hybrid systems. In *Proceedings of the 42nd IEEE Conference on Decision and Control*, volume 1, pages 167–172, December 2003.

[83] B. Wahlberg, H. Hjalmarsson, and J. Mårtensson. Variance results for identification of cascade systems. *Automatica*, 45(6):1443–1448, 2009.

[84] J. Wang, W. X. Zheng, and T. Chen. Identification of linear dynamic systems operating in a networked environment. *Automatica*, 45(12):2763–2772, Dec. 2009.

[85] Jiantao Wang, David X. Wei, and Steven H. Low. Modelling and stability of FAST TCP. In *Proceedings of IEEE Infocom 2005*, March 2005.

[86] L. Y. Wang and G. G. Yin. Asymptotically efficient parameter estimation using quantized output observations. *Automatica*, 43(7):1178–1191, Jul. 2007.

[87] L. Y. Wang and G. G. Yin. Quantized identification with dependent noise and Fisher information ratio of communication channels. *IEEE Trans. Aut. Control*, 55(3):674–690, Mar. 2010.

[88] L. Y. Wang, G. G. Yin, J.-F. Zhang, and Y. Zhao. *System identification with quantized observations*. Birkhäuser, Boston, USA, 2010.

[89] C. J. C. H. Watkins and P. Dayan. Q-learning. *Mach. Learn.*, 8:279–292, 1992.

[90] David X. Wei. Congestion control algorithms for high speed long distance TCP connections. Master's thesis, Caltech, 2004.

[91] P.J. Werbos. A menu of designs for reinforcement learning over time. In W.T. Miller, R.S. Sutton, and P.J. Werbos, editors, *Neural Networks for Control*, pages 67–95. MIT Press, Cambridge, MA, 1991.

[92] A. Wills and Lennart Ljung. Wiener system identification using the maximum likelihood method. In F. Giri and E. W. Bai, editors, *Block-Oriented Nonlinear System Identification*, number Lecture Notes in Control and Information Science no 404. Springer, September 2010.

[93] A. Wills, T. B. Schön, L. Ljung, and B. Ninness. Blind identification of Wiener models. In *Proc. IFAC Congress*, Milan, Italy, Sept. 2011.

[94] J. J. Xiao, A. Ribeiro, Z. Q. Luo, and G. B. Giannakis. Distributed compression-estimation using wireless sensor networks - The design goals of performance, bandwidth efficiency, scalability, and robustness. *IEEE Sig. Proc. Mag.*, 23(4):27–41, jul 2006.

[95] L. Xiao, S. Boyd, and S. Lall. A scheme for robust distributed sensor fusion based on average consensus. In *Proceedings of International Conference on Information Processing in Sensor Networks*, pages 67–70, Los Angeles, CA, USA, April 2005.

[96] Xin Yang, Haoying Fu, Hongyuan Zha, and Jesse Barlow. Semi-supervised nonlinear dimensionality reduction. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pages 1065–1072, New York, NY, USA, 2006. ACM.

[97] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society, Series B*, 68:49–67, 2006.
[98] L. A. Zadeh. On the identification problem. *IRE Transactions on Circuit Theory*, 3:277–281, 1956.
[99] Z. Zang, R. R. Bitmead, and M. Gevers. Iterative weighted least-squares identification and weighted LQG control design. *Automatica*, 31(11):1577–1594, 1995.

APPENDIX

DETAILS OF RL FOR THE LQ PROBLEM

The basic HDP algorithm is given by

i) Take $V_0(x) = 0$ and set $j = 0$.

ii) Solve for $u_j(x)$ as follows:

$$u_j(x) = \arg\min_u x^T Qx + u^2 + V_j(Ax + Bu) \quad (76)$$

iii) Perform the update

$$V_{j+1}(x) = x^T Qx + u_j^2(x) + V_j(Ax + Bu_j(x)) \quad (77)$$

iv) Set $j \to j + 1$ and goto Step ii)

For our linear problem the Steps ii) and iii) become the corresponding steps in Section V-D.

To obtain a version which does not require knowledge of the state transition matrix A substitute $V_j(x)$ and $u_j(x)$ for parametrized approximations $\hat{V}(x, w_j)$ and $\hat{u}(x, q_j)$, where $w_j \in \mathbb{R}^n$ and $q_j \in \mathbb{R}^m$, for some positive integers n and m. Typically neural networks are used as function approximators. In each iteration the parameters are updated using measurements. The details of one possible algorithm are as follows. In iteration j, N state measurements $\{x(jN+k)\}_{k=1}^N$ using the most recent control policy are collected.

i') Set w_0 such that $\hat{V}(x, w_o) = 0$ and let q_0 correspond to an initial control policy. Let the initial state of the system be x_0. Set $j = 0$.

iia') Perform a sequence of control actions $\hat{u}(x(jN+k), q_j)$, $k = 0, \ldots N-1$ resulting in new system states $\{x(jN+k+1)\}$.

iib') Set

$$\begin{aligned}&\hat{V}_j^+(x(jN+k), w_j, q) = \\ &x(jN+k)^T Qx(jN+k) + \hat{u}^2(x(jN+k), q) \\ &+ \hat{V}(Ax(jN+k) + B\hat{u}(x(jN+k), q), w_j). \end{aligned} \quad (78)$$

iic') Update the control parameters using one step in a gradient descent algorithm aiming at decreasing $\sum_{k=0}^{N-1} \hat{V}_j^+(x(jN+k), w_j, q)$:

$$q_{j+1} = q_j - \gamma \sum_{k=0}^{N-1} \frac{\partial \hat{V}_j^+(x(jN+k), w_j, q)}{\partial q}\Big|_{q=q_j} \quad (79)$$

where $\gamma > 0$ is the step-size of the update.

iii') Solve for

$$\begin{aligned} w_{j+1} = \arg\min_w \sum_{k=0}^{N-1} \Big(&\hat{V}(x(jN+k), w) - \\ &x^T(jN+k)Qx(jN+k) - \hat{u}^2(x(jN+k), K_j) \\ &-\hat{V}(x(jN+k+1), w_j)\Big)^2. \end{aligned} \quad (80)$$

iv') Set $j \to j + 1$ and goto Step iia').

For our LQ problem we can use $\hat{V}(x, w) = x^T Px$, where w is a complete linear parametrization of $P = P^T$, and $q = K$ so that $\hat{u}(x, q) = -Kx$. The P corresponding to w_j is denoted P_j. Then

$$\begin{aligned}&\hat{V}_j^+(x_j, w_j, q) = \\ &x_j^T Qx_j + x_j^T K^T Kx_j + x_j^T (A - BK)^T P_j (A - BK)x_j\end{aligned}$$

and

$$\begin{aligned}\frac{\partial \hat{V}_j^+(x_j, w_j, q)}{\partial q}\Big|_{q=q_j} &= 2(Kx_jx_j - 2x_j^T(A - BK)^T P_jBx_j) \\ &= 2(Kx_jx_j - 2((x_j^{(0)})^T P_jBx_j).\end{aligned}$$

Notice that this partial derivative can be evaluated without explicit knowledge of A. The necessary information regarding A is contained in the next state $x_j^{(0)}$. We also notice that iii') gives exactly iii) in Section V-D if $\big[x(jN), \ldots, x(jN+N-1)\big]$ has full rank. In conclusion the algorithm above only uses explicit knowledge of B; pertinent information regarding A is contained in measurements. There are other algorithms where explicit knowledge of B can be avoided, e.g. Q-learning [89].

OPTIMAL EXPERIMENT DESIGN IN EXAMPLE 8

Let $\hat{A}$ and $\hat{B}$ be prediction error estimates of A and B; note that since the measurement equation (73) does not contain any unknown parameters, the basis for the state space is well defined and all elements of the system matrices are identifiable. Consider now the following stochastic open loop optimal experiment design problem: The estimate of $J_o(x_o)$ is given by $x_o^T P(\hat{A}, \hat{B}, \bar{K})x_o$. Form the quality measure

$$V(\hat{A}, \hat{B}) = (x_o^T P(\hat{A}, \hat{B}, \bar{K})x_o - J_o(x_o))^2. \quad (81)$$

Let N be the length of the identification experiment. The optimal open loop experiment design problem we are interested in is then given by

$$\begin{aligned}&\min_{u_{id}(0), u_{id}(1), \ldots, u_{id}(N)} V(\hat{A}, \hat{B}) \\ &\sum_{t=1}^{N} x_{id}^T(t)Qx_{id}(t) + u_{id}^2(t) \le J_o(x_o)\end{aligned} \quad (82)$$

(u_{id} and x_{id} are defined as in Example 7). The constraint imposes that the experimental cost can not be larger than the LQG cost (69). As this problem is not computationally tractable, e.g. the cost is a random quantity, we will approximate this by the techniques in [41]; in particular we assume the input to be stationary. This leads to the following formal problem

$$\begin{aligned}&\min_{\Phi_u} \quad \mathrm{E}[V(\hat{A}, \hat{B})] \\ &N\mathrm{E}\left[x_{id}^T(t)Qx_{id}(t) + u_{id}^2(t)\right] \le J_o(x_o)\end{aligned} \quad (83)$$

where Φ_u is the spectrum of the input. The problem (83) corresponds to an 'on average' approximation of (82). By way of second order approximations, this problem can be converted into a semi-definite program [41].