Expressivity of hidden Markov chains vs. Recurrent neural networks from a system theoretic viewpoint

François Desbouvries, Yohan Petetin, Achille Salaün

Expressivity of Hidden Markov Chains vs. Recurrent Neural Networks from a system theoretic viewpoint

François Desbouvries, *Senior Member, IEEE*, Yohan Petetin, *Member, IEEE*, and Achille Salaün

***Abstract*—Hidden Markov Chains (HMC) and Recurrent Neural Networks (RNN) are two well known tools for predicting time series. Even though these solutions were developed independently in distinct communities, they share some similarities when considered as probabilistic structures. So in this paper we first consider HMC and RNN as generative models, and we embed both structures in a common generative unified model (GUM). We next address a comparative study of the expressivity of these models. To that end we assume that the models are furthermore linear and Gaussian. The probability distributions produced by these models are characterized by structured covariance series, and as a consequence expressivity reduces to comparing sets of structured covariance series, which enables us to call for stochastic realization theory (SRT). We finally provide conditions under which a given covariance series can be realized by a GUM, an HMC or an RNN.**

***Index Terms*—Hidden Markov Chains, Recurrent Neural Networks, Generative Models, Expressivity, Modeling Power, Stochastic Realization Theory.**

I. Introduction

Let $x_{0:t} = (x_0, \cdots, x_t)$ be a sequence of random variables (r.v.). We focus on the general problem of predicting a future observation x_{t+1} from a realisation of $x_{0:t} = (x_0, \cdots, x_t)$. This problem has many applications such as speech recognition, finance or geology [1][2] and can be addressed through Bayesian estimation in two ways. The first way consists in estimating a generative model $p_\theta(x_{0:t})$, for all $t \in \mathbb{N}$, and next computing the posterior distribution $p_\theta(x_{t+1}|x_{0:t})$. The second approach aims at building directly a function f_θ such that $f_\theta(x_{0:t})$ is close to x_{t+1} is a given sense. The objective of this paper is to propose a comparison between two key tools associated with each approach, hidden Markov Chains (HMC) on the one hand, and recurrent neural architectures (RNN) on the other hand. Our study is *not* of an experimental nature (see e.g. [3][4] for such comparisons), but rather aims at quantifying the modeling power of each model. Before further comparing these two models, let us briefly review the rationale of the two approaches by recalling the prediction problem in the static case.

François Desbouvries and Yohan Petetin are with Samovar, Telecom SudParis, Institut Polytechnique de Paris, Evry, France (e-mail: francois.desbouvries,yohan.petetin@telecom-sudparis.eu). Achille Salaün is with Institute of Biomedical Engineering, Department of Engineering Science, University of Oxford, Oxford, UK. The work was performed when he was with Samovar, Telecom SudParis, Institut Polytechnique de Paris, Evry, France, and Nokia Bell Labs, Nozay, France (e-mail: achille.salaun@eng.ox.ac.uk).

A. Bayesian problem

Let us consider a sample (x, y) from a joint probability density function (pdf) $p(x, y)$. The objective is to predict y from x, so we look for an estimator $\hat{y} = f(x)$ such that $\hat{y}$ is "close" to y. In a Bayesian context, building estimator $f(.)$ is induced by the choice of a loss function $L(.,.)$, which depends on the problem at hand, and quantifies the error between the prediction $f(x) = \hat{y}$ and the true variable y. Building the associate estimator amounts to minimizing the Bayesian risk

$$R(f) = \mathbb{E}\left[L(f(x), y)\right], \tag{1}$$

i.e. build $f^\star(x) = \hat{y}$ in which $f^\star = \underset{f}{\operatorname{argmin}}\, R(f)$. One can show that $f^\star$ depends on the *posterior* density $p(y|x) = \frac{p(x,y)}{p(x)}$ (also called *predictive distribution* in the context of prediction). For instance, if the loss is quadratic, the Bayesian estimator is well known to be the conditional expectation, $f^\star(x) = \hat{y} = \mathbb{E}[y|x]$. However, $p(x, y)$ is not known in practice. To cope with this problem one can estimate the Bayesian risk by two different ways: by introducing a parameterized distribution $p_\theta(x, y)$, or by estimating integral (1) from Monte Carlo samples.

1) Parameterizing the joint distribution $p(x, y)$: The first approach consists in proposing a model of the unknown pdf p. We thus restrict ourselves to a parameterized set of pdfs $(p_\theta)_{\theta\in\Theta}$, in which θ can be multidimensional. If we have a set of labelled independent samples

$$\mathcal{E} = \left\{(x_i, y_i) \overset{\text{i.i.d.}}{\sim} p(x, y)\right\}_{1\le i\le n}, \tag{2}$$

the relevance of p_θ can be quantified via the likelihood function [5], [6]

$$\mathcal{L}(.; \mathcal{E}) : \theta \mapsto \prod_{i=1}^{n} p_\theta(x_i, y_i), \tag{3}$$

so approximating p amounts to computing a parameter θ which maximizes the likelihood. Note however that the choice of the parametric family is critical: p_θ should model the data at hand, and in the same time function (3) should be computed and optimized efficiently. In general, maximizing the likelihood can only be done approximately. For instance in the case of latent variables models (i.e. the model is defined through the introduction of an unobserved random variable h such that $p_\theta(x, y) = \int p_\theta(h, x, y)\mathrm{d}h$), maximizing the likelihood requires approximating schemes such as the Expectation Maximization (EM) algorithm [7].

2) Parameterizing the estimator $f(x)$: The second approach does not make any assumption on $p(x,y)$, but rather estimates (1) from the same dataset (2) (see e.g. [8], [9]). The problem of building an estimator becomes that of minimizing the empirical risk

$$f_n^\star = \underset{f}{\operatorname{argmin}} \frac{1}{n}\sum_{i=1}^{n} L(f(x_i), y_i). \tag{4}$$

However, since the dataset is finite, in the absence of further constraints, any function interpolating the points (x_i, y_i) satisfies the optimisation problem (4). In such a case, the model overfits and proves unable to generalize to new observations. This problem is often overcome by chosing a family of functions $(f_\theta)_{\theta\in\Theta}$, and finally (4) turns into the parameter estimation problem:

$$\theta_n^\star = \underset{\theta}{\operatorname{argmin}}\ R_n(f_\theta), \tag{5}$$

$$R_n(f_\theta) = \frac{1}{n}\sum_{i=1}^{n} L(f_\theta(x_i), y_i) \tag{6}$$

which eventually produces the estimator $\hat{y} = f_{\theta_n^\star}(x)$ (notation $\theta_n^\star$ underlines the fact that the estimator depends on the training set $\mathcal{E}$, which is of dimension n).

As above, the choice of the family $(f_\theta)_{\theta\in\Theta}$ should be balanced: a poor set of functions will lead to unrealistic predictions, while a rich set of functions can lead to overfitting. Moreover $(f_\theta)_{\theta\in\Theta}$ should lead to tractable learning, i.e. it should be possible to solve (5) efficiently. Classical solutions include the functions belonging to a reproducing kernel Hilbert space (RKHS) [10] [11] and the functions defined by a neural network (NN) [12] [13]. Optimizing (5) for these families of functions leads to well known algorithms such as (linear or kernel based) least squares [8], Support Vector Machines (SVM) [14] [15] [16], or deep learning algorithms [8] [17] for regression or classification.

3) Discussion: As we have just seen, for minimizing (5) it is not necessary to mimic the distribution $p(x,y)$ by $p_\theta(x,y)$. In addition, under some assumptions about the family $(f_\theta)_{\theta\in\Theta}$ it is possible to derive concentration inequalities such as

$$\mathbb{P}\left(|R_n(f_{\theta_n^\star}) - R(f_{\theta_n^\star})| > \epsilon\right) \leq \delta_{\epsilon,n},$$

where $\delta_{\epsilon,n} \to 0$ when $n \to \infty$ [18]; in other words, such a bound ensures that $f_{\theta_n^\star}$ generalizes well and also provides a rate of convergence. However, note that in some contexts, and in particular for times series analysis, we may be interested in predicting $\phi(x)$ from y for a large class of functions ϕ; once the joint distribution has been estimated, it is possible to comply with such a constraint without running a new estimation algorithm for each function ϕ. In addition, the knowledge of the posterior distribution enables to quantify (even approximately) the uncertainty of the prediction.

B. Goal of this paper

The two previous approaches can be adapted to the sequential constraints induced by time series analysis. Let t be the current time parameter. In order to represent the joint distribution of $x_{0:t}$, we need to choose a parametric generative model $p_\theta(x_{0:t})$ such that θ does not depend on $t \in \mathbb{N}$ (otherwise, the model cannot be used with new observations). p_θ should model the time series $x_{0:t}$ in a realistic way, and so take into account the dependencies between the observations; in the same time, we should be able to compute an estimator of θ and to approximate the posterior distribution $p_\theta(x_{t+1}|x_{0:t})$ for any realization $x_{0:t}$.

A popular model satisfying these requirements is the HMC, particularly developed in the signal processing community, see e.g. [19] [20]. In the same way, it is possible to parameterize a function $f_\theta(x_{0:t})$ in a such way that θ does not depend on t. This is the principle of RNN particularly developed in the machine learning community [21]. Even if such models were basically proposed for point estimation, they can be easily used for building generative models. So from now on, in order to compare the two approaches in a common framework, we will consider that we have at our disposal two generative models $p_\theta(x_{0:t})$ for all t, the HMC and the RNN. Starting from the observation that both models actually rely on a set of latent variables and that they share some common features in the construction of these variables, our objective in this paper is to quantify thoroughly (under some assumptions) how the structural differences of these models impact on their expressivity.

The rest of this paper is organized as follows. In Section II we start by formalizing both models under a common framework. Next in section III, we see that comparing both models under the linear and Gaussian stationary assumptions reduces to comparing the covariance series of the stochastic process $x_{0:t}$, and consequently the study calls on Stochastic Realization Theory (SRT) (a branch of systems theory). Section IV provides a brief summary of SRT. Finally the expressivity of HMC and RNN is compared in section V.

II. Latent data models for time series analysis

In this section we introduce our two generative models based on a sequence of latent r.v. $h_{0:t}$, and we next embed them as two particular instances of a more general model.

A. Markovian models

When dealing with a time series $x_{0:t}$ one major issue consists in modeling the dependency between the observations. For example, a simple Markov Chain (MC)

$$p_\theta(x_{0:t}) = p_\theta(x_0)\prod_{s=0}^{t-1} p_\theta(x_{s+1}|x_s) \tag{7}$$

is often unlikely to represent the distribution of $x_{0:t}$ in a realistic way , since x'_t, $t' < t-1$, becomes independent of x_t when x_{t-1} is observed. One way of enhancing expressivity is to introduce a latent process $h_{0:t}$, where each h_s can be discrete or continuous. The model is now described by the full joint density $p(h_{0:t}, x_{0:t})$, from which $p(x_{0:t})$ is obtained by marginalizing out the latent variables. This marginalization definitely makes pdf $p(x_{0:t})$ more complex and so increases the modeling power. A constraint is that the introduction of a latent process should preserve some computational properties

in order to be used in practice. In this sense, the HMC generalizes (7) by adding a latent process in a rather parsimonious way, since its joint pdf reads:

$$p_\theta(h_{0:t}, x_{0:t}) \overset{\text{HMC}}{=} p_\theta(h_0) \prod_{s=1}^{t} p_\theta(h_s|h_{s-1}) \prod_{s=0}^{t} p_\theta(x_s|h_s). \quad (8)$$

So an HMC benefits of three (conditional) independence properties: the latent process is an MC; given the latent variables $h_{0:t}$, observations $x_{0:t}$ are independent; and given all latent variables $h_{0:t}$, an observation only depend on the latent variable at the same time, $p_\theta(x_s|h_{0:t}) = p_\theta(x_s|h_s)$ for all s, $0 \leq s \leq t$. The hidden process $h_{0:t}$ can have a physical meaning (in which case estimating $h_{0:t}$ from $x_{0:t}$ is relevant). If not, the role of the latent variables $h_{0:t}$ is just to make the observed process $x_{0:t}$ more complex, and the HMC can be seen as a generative model. Associated inference algorithms for approximating the Maximum Likelihood estimator and posterior distributions have been extensively studied for these models [22] [2] and are recalled in Appendix A.

B. RNN architectures

RNN are an adaptation of neural architectures to times series. So let us start by briefly recalling the rationale of neural networks.

Neural network architectures are versatile classes of functions [17], which have found many applications for classification or prediction, including language [23] or image [24] processing. A neural network (NN) is a succession of parameterized functions called neurons. A neuron typically computes $\boldsymbol{x} \mapsto \sigma(\boldsymbol{w}\boldsymbol{x} + b)$, where $\boldsymbol{w}\boldsymbol{x}$ is the dot product of $\boldsymbol{w}$ (a vector of weights) and $\boldsymbol{x}$ (a vector of variables), b is the bias, and $\sigma(.)$ is a so-called (nonlinear) activation function, such as the sigmoid, hyperbolic tangent or ReLu functions. Neurons can be gathered into layers which themselves can be cascaded, yielding increasingly complex functions. Some universal approximation theorems have been proposed [25], [26], [27], [28]; for instance, given any (possibly multi-dimensionnal) continuous function f, there exists a single-layer NN f_θ arbitrarily close to f, provided the activation function is not polynomial [27]. Similar results have been proposed for multiple layers NNs. So any Lebesgue-integrable function $f : \mathbb{R}^n \to \mathbb{R}$ can be approximated by an NN with ReLu activation function and layers made of at least $n + 4$ neurons, provided the net is deep enough [28]. The number of layers and of neurons per layer, as well as the activation functions, are hyperparameters which characterize the NN architecture, and the weights and biases are the model parameters learnt from a training set. However, the input of an NN as described above is of fixed size, which is not well suited to the modeling of time series in which observations accumulate - unless we use a sliding window, but in that case the prediction would not be based on the full set of observations.

In order to introduce dependencies between all the observations, RNN introduce a latent variable h_t which is a function of all observations $x_{0:t}$ and serves as a memory of the past. After receiving the new information x_t, the new state is computed as

$$h_t = f_\theta(h_{t-1}, x_t), \quad h_{-1} = 0 \quad (9)$$

where θ is an NN layer. In other words, h_t is a summary of all the past observations until time t. Finally a prediction of x_{t+1} is computed as $\hat{x}_{t+1} = g_\theta(h_t)$ where g_θ is an NN architecture. As we claimed before, RNN can be transformed into generative models by replacing function g_θ by a parametric distribution $p_\theta(x_{t+1}|h_t)$. In this case, we obtain a family of models defined by

$$p_\theta(x_{0:t}) \overset{\text{RNN}}{=} p_\theta(x_0) \prod_{s=0}^{t-1} p_\theta(x_{s+1}|h_s), \quad (10)$$

where h_s depends on θ from (9).

By construction, the posterior distribution $p_\theta(x_{t+1}|x_{0:t})$ coincides with $p_\theta(x_{t+1}|h_t)$ and is directly available. The estimation of θ can be computed by a gradient backpropagation algorithm [29] [30] [31] [32] which aims at maximizing $\log(p_\theta(x_{0:t}))$ for a given observation. Due to the time component, there can be as many computed gradients as observations for a given parameter.

However, in practice, the gradients computed for a given parameter geometrically tend to infinity or to zero when we get back into the past. These phenomena are called exploding gradient and vanishing gradient. The exploding gradient phenomenon is often due to the repeated multiplication of high weights, a situation where learning the RNN becomes particularly unstable. An efficient way to limit this behavior is to bound the values taken by the gradient [17], [33]. One can also include a regularization term to the cost function in order to penalize weights that are too large [34]. By contrast, the vanishing gradient phenomenon results from the repeated multiplication or small size weights, as well as the iterated use of activation functions which have derivatives bounded by 1 in magnitude (*e.g.* the sigmoid). In that case, the oldest observations are not taken into account in the learning phase, so it is difficult to learn long term dependencies. In order to mitigate the vanishing gradient phenomenon, more sophisticated architectures have been proposed, such as the Long Short Term Memories (LSTM) and the Gated Recurrent Units (GRU) [35] (the only difference is that the corresponding parameterization of f_θ becomes more complex).

C. Generative unified model

As we have just seen, HMC and RNN models result from a different paradigm but both aim at proposing a parameterized distribution $p_\theta(x_{0:t})$ via the introduction of latent variables $h_{0:t}$. In the RNN model, $h_{0:t}$ is deterministic given the observations $x_{0:t}$, and h_s summarizes all the observations up to time s into a unique variable; in the HMC model, $h_{0:t}$ is stochastic given the observations, and indeed the Bayesian estimation of $h_{0:t}$ is of interest in cases where $h_{0:t}$ is a physical process of interest.

When we put aside the computational aspects, the natural question that arises is to compare the set of distributions $p_\theta(x_{0:t})$ induced by each model. Actually, both representations

can be reconciled as particular instances of the following Generative Unified Model (GUM),

$$p_\theta(h_{0:t}, x_{0:t}) \overset{\text{GUM}}{=} p_\theta(h_0) \prod_{s=1}^{t} p_\theta(h_s|h_{s-1}, x_{s-1}) \prod_{s=0}^{t} p_\theta(x_s|h_s). \tag{11}$$

Indeed, the HMC model (8) is a GUM where

$$p_\theta(h_t|h_{t-1}, x_{t-1}) \overset{\text{HMC}}{=} p_\theta(h_t|h_{t-1}), \tag{12}$$

while the RNN (9)- (10) satisfies (up to the transformation $h_t \leftarrow h_{t-1}$)

$$\begin{aligned} p_\theta(h_t|h_{t-1}, x_{t-1}) &\overset{\text{RNN}}{=} \delta_{f_\theta(h_{t-1}, x_{t-1})}(h_t), \\ p_\theta(x_0|h_0) &\overset{\text{RNN}}{=} p_\theta(x_0), \\ h_0 &\overset{\text{RNN}}{=} 0, \end{aligned} \tag{13}$$

where δ denotes the Dirac mass. In the rest of this paper we will also consider deterministic GUM (D-GUM), which are defined by the first equation of (13) only (the interest of D-GUM over RNN will be clear in section V-B).

Let us now discuss the three models, beginning with their similarities. First, in all three models the pair (h_s, x_s) is an MC:

$$\begin{aligned} p(h_s, x_s|h_{0:s-1}, x_{0:s-1}) &= p(h_s, x_s|h_{s-1}, x_{s-1}) \\ &= p(h_s|h_{s-1}, x_{s-1}) p(x_s|h_s), \end{aligned} \tag{14}$$

which induces $p(x_s|h_{0:s}, x_{0:s-1}) = p(x_s|h_s)$. In addition, in all three models the marginal process $h_{0:t}$ is also an MC, since

$$\begin{aligned} p_\theta(h_s|h_{0:s-1}) &= \int p_\theta(h_s|h_{s-1}, x_{s-1}) p_\theta(x_{s-1}|h_{s-1}) \mathrm{d}x_{s-1} \\ &= p_\theta(h_s|h_{s-1}). \end{aligned}$$

As a result, the GUM, HMC and RNN models only differ via the distribution $p_\theta(x_{0:t}|h_{0:t})$. In an HMC, h_s only depends on h_{s-1} given the past $(h_{0:s-1}, x_{0:s-1})$, so $p_\theta(x_{0:t}|h_{0:t}) = \prod_{s=0}^{t} p(x_s|h_s)$; on the other hand, h_t is stochastic so $p_\theta(x_{0:t})$ is not available in closed form. By contrast, in a D-GUM or an RNN, h_s is deterministic given the past $(h_{0:s-1}, x_{0:s-1})$, so $p_\theta(x_{0:t})$ is available in closed form; but h_s also depends on h_{s-1} so $p_\theta(x_{0:t}|h_{0:t})$ is difficult to interpret. The graphical representation of the three models is displayed in Fig. 1.

Now that we have cast the HMC and the RNN in a common framework, we can address the comparison of their expressivity from the GUM perspective. More precisely, our objective is to set some assumptions which enable us to discuss on the distribution of the observation $p_\theta(x_{0:t})$ associated to the GUM, and next to discuss on the restrictions on this distribution induced by (12) and (13).

III. Structure of the mapping $\theta \to p_\theta(x_{0:t})$: the linear and Gaussian case

A. Linear and Gaussian GUM

We now address the expressivity of the HMC and RNN models. In order to compare the observations pdf $p(x_{0:t})$ induced by the HMC and RNN models, we set ourselves in the general framework of GUM, in which $p(x_{0:t})$ is a marginal of (11). Of course, $p(x_{0:t}) = \int_{h_{0:t}} p(h_{0:t}, x_{0:t}) \mathrm{d}h_{0:t}$ cannot, in most cases, be computed in closed form. In order to be able to provide a compared analysis of the expressivity of those models (and thus to understand, when we reduce to the particular cases of HMC and of RNN, the role of stochastic vs. deterministic transitions: see (12) and (13)), we consider the simplified linear and Gaussian framework, i.e. a GUM model in which the elementary factors in (11) read

$$p(h_0) = \mathcal{N}(h_0; 0; \eta), \tag{15}$$

$$p(h_t|h_{t-1}, x_{t-1}) = \mathcal{N}(h_t; a h_{t-1} + c x_{t-1}; \alpha), \tag{16}$$

$$p(x_t|h_t) = \mathcal{N}(x_t; b h_t; \beta), \tag{17}$$

in which h_t is an n-dimensional vector and x_t is a scalar; so a, b and c are respectively $n \times n$, $1 \times n$, $n \times 1$, η and α are $n \times n$ covariance matrices, and $\beta \geq 0$. $\theta = (a, b, c, \alpha, \beta, \eta)$ is the parameter of the model.

It is easy to see that in model (11) (15)-(17), the joint pdf $p(x_{0:t})$ is a zero-mean multivariate Gaussian density, which is fully characterized by its covariance matrix. Let η_t be the covariance matrix of h_t (we will later see how η_t depends on η_0 and on time t). We have $\mathrm{Var}(x_t) = \beta + b\eta_t b^T$, and, for all $t \in \mathbb{N}$, $k \in \mathbb{N}^*$,

$$\mathrm{Cov}(x_t, x_{t+k}) = b(a+cb)^{k-1} \underbrace{(a\eta_t b^T + c(\beta + b\eta_t b^T))}_{N_t}. \tag{18}$$

Due to η_t and factor N_t, $\mathrm{Var}(x_t)$ and $\mathrm{Cov}(x_t, x_{t+k})$ *a priori* depend on time t. In order to simplify the analysis (see §V below) we first look for simple sufficient conditions yielding stationarity.

B. Stationnarity

First, it is easy to see that the matrix series $(\eta_t)_{t\in\mathbb{N}}$ is defined by

$$\eta_{t+1} = (\alpha + c\beta c^T) + (a + cb)\eta_t(a + cb)^T. \tag{19}$$

As a consequence, for all $t \in \mathbb{N}$, $\eta_{t+1} - \eta_t = (a + cb)^t [\eta_1 - \eta_0] (a + cb)^{tT}$. The series (η_t) is thus constant if

$$\eta_0 = \eta_1 = \eta. \tag{20}$$

Assumption (20) implies in turn that $\mathrm{Var}(x_t)$ and $\mathrm{Cov}(x_t, x_{t+k})$ no longer depend on time, so that $(x_t)_{t\in\mathbb{N}}$ is a wide sense stationnary process. Let us finally remark that under assumption (20), equation (19) becomes:

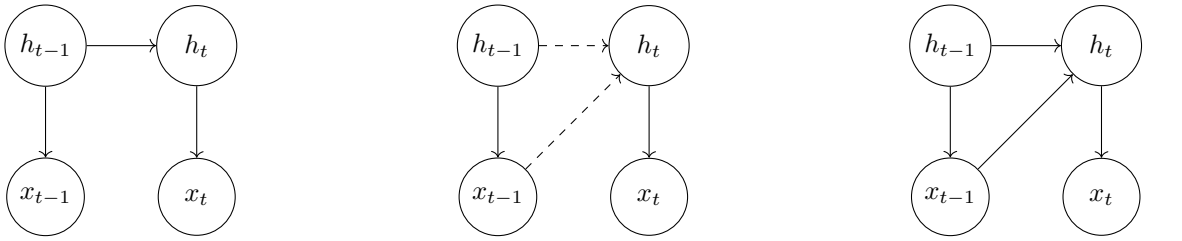
$$\eta = (\alpha + c\beta c^T) + (a + cb)\ \eta\ (a + cb)^T. \tag{21}$$

This equation in variable η is meaningfull (recall that η is a covariance matrix) only if it admits a semi-definite positive (≥ 0) solution, which implies [36], [37] that

$$(a + cb) \text{ has all its eigenvalues in } \{z \in \mathbb{C}; |z| < 1\}. \tag{22}$$

We will now assume that (22) and (20) hold, which implies that x_t is stationary. Let us note that this stationarity assumption is reasonable, because under assumption (22), the series (η_t) converges when t tends to infinity. So $(h_t)_{t\in\mathbb{N}}$ as well as $(x_t)_{t\in\mathbb{N}}$ are at least asymptotically stationary.

Remark 1: In the HMC case, we have the additional constraint $c = 0$ since h_t does not depend on x_{t-1} given (h_{t-1}, x_{t-1}). In the case of D-GUM, $\alpha = 0$; if we also



(a) HMC (b) RNN (c) GUM

Fig. 1: Conditional dependencies in HMC, RNN, and GUM. The dashed (resp. solid) lines stand for deterministic (resp. probabilistic) dependencies.

consider the full RNN case with its particular initial conditions (see (13)), the constraint $\mathrm{Var}(h_1) = \eta = c\mathrm{Var}(x_0)c^T = c(\beta + b\eta b^T)c^T$ has also to be satisfied.

C. The mapping $\theta \to p_\theta(x_{0:t})$

Let $r_k = \mathrm{Cov}(x_t, x_{t+k})$. We observe that this covariance series has a very specific structure:

$$r_0 = \beta + b\,\eta\,b^T; \tag{23}$$

$$\text{for all } k \in \mathbb{N}^*,$$

$$r_k = \underbrace{b}_{H}\underbrace{(a+cb)}_{F}^{k-1}\underbrace{(a\eta b^T + c(\beta + b\eta b^T))}_{N}. \tag{24}$$

Since this covariance series $(r_k)_{k\in\mathbb{N}}$ characterizes the distribution of $p_\theta(x_{0:t})$ for all t, we now consider function

$$\phi : \underbrace{(a, b, c, \alpha, \beta, \eta)}_{\theta} \overset{(23)-(24)}{\longrightarrow} \phi(\theta) = (r_k)_{k\in\mathbb{N}}, \tag{25}$$

in which $(r_k)_{k\in\mathbb{N}}$ is given by (23) (24). Since a study of the direct range of ϕ under the HMC or RNN constraints seems a difficult task, we rather consider the inverse mapping.

Let us first observe that the factorized structure of the covariance series (i.e., there exists (H, F, N) s.t. $r_k = HF^{k-1}N$ for all $k \geq 1$) is remarkable, and is directly related to system theory. More precisely, the output of any linear time invariant (LTI) state space system has a stationary factorized covariance series, and conversely, any such series can be *realized* by an LTI system, This second point is the topic of *Stochastic realization theory (SRT)*, which indeed is of interest here since we shall look for parameters $\theta = (a, b, c, \alpha, \beta, \eta)$ s.t. $\phi(\theta) = (r_k)_{k\in\mathbb{N}}$ for a given $(r_k)_{k\in\mathbb{N}}$. Before we proceed to the analysis we thus briefly provide a brief reminder of SRT (the reader familiar with SRT can skip section IV and directly jump to section V).

IV. A SHORT REVIEW OF SRT

Let us briefly review some points from SRT [38], [36], [39], [40], [41] which we will need in section V. SRT is a part of systems theory, which deals with modeling, controlling and estimating dynamic systems (see e.g. [42], [43], [44]). Before we proceed (see section IV-B) we need to recall some algebraic facts from deterministic realization theory (DR).

A. DRT

Let us consider a linear discrete time system with state h_t:

$$\begin{cases} h_{t+1} &= Fh_t + Nu_t \\ x_t &= Hh_t \end{cases}, \tag{26}$$

where F (resp. N, H) are $n \times n$ (resp. $n \times 1$, $1 \times n$) matrices (we only deal here with the case where observation x_t and input u_t are one-dimensional). The mapping between input u_t and output x_t is given by the convolution equation $x_t = \sum_{k=1}^{+\infty} H_k u_{t-k}$, where the lags H_k of the impulse response (the so-called Markov parameters of the system) satisfy

$$H_k = HF^{k-1}N \tag{27}$$

for all $k \geq 1$. Equivalently, the strictly causal transfer function $H(z) = \sum_{k=1}^{+\infty} H_k z^{-k}$ can be written as $H(z) = H(zI - F)^{-1}N$.

The DR problem consists in building three matrices H, F, N, with $F_{n\times n}$ of minimal dimension, from the impulse response of the system, i.e. move from the infinite representation $(H_k)_{k\in\mathbb{N}^*}$ to the finite representation (H, F, N), with F of minimal dimension. The key tool for this problem is the infinite Hankel matrix

$$\mathcal{H}_\infty = \begin{bmatrix} H_1 & H_2 & H_3 & \dots \\ H_2 & H_3 & & \\ H_3 & & & \\ \vdots & & & \end{bmatrix}. \tag{28}$$

From (27), $\mathcal{H}_\infty$ factorizes as

$$\mathcal{H}_\infty = \begin{bmatrix} H \\ HF \\ HF^2 \\ \vdots \end{bmatrix} .[N, FN, F^2N, ...], \tag{29}$$

and so has finite rank, which moreover is equal to n (the dimension of F) if and only if (iff.) each factor is itself full rank n. Conversely, if $\mathcal{H}_\infty$ has finite rank n, then it can be factorized as a product of two factors of dimensions $(\infty \times n)$ and $(n \times \infty)$, both of them being of full rank n, and due to the Hankel structure, there exists $F_{n\times n}, N_{n\times 1}, H_{1\times n}$ so that (29) (and thus (27)) is satisfied. Moreover, from the proposition below, all minimal realizations of $H(z)$ are isomorphic:

Proposition 1: [45, proposition 3] (H_1, F_1, N_1) and (H_2, F_2, N_2) are two minimal realizations of $H(z)$ if and

only if there exists T invertible such that $F_2 = TF_1T^{-1}$, $N_2 = TN_1$ and $H_2 = H_1T^{-1}$.
Finally numerically efficient DR algorithms have been proposed in [46], [47].

B. SRT

Let us now consider the state space system

$$\begin{cases} h_{t+1} &= Fh_t + u_t \\ x_t &= Hh_t + v_t \end{cases}, \tag{30}$$

where h_0 is zero-mean and uncorrelated with (u_t, v_t), and where (u_t, v_t) is a zero-mean, uncorrelated, stationary random process with

$$\mathbb{E}\left[\begin{bmatrix} u_t \\ v_t \end{bmatrix} . \begin{bmatrix} u_{t'}^T & v_{t'}^T \end{bmatrix}\right] = \begin{bmatrix} Q & S \\ S^T & R \end{bmatrix} \delta_{t,t'} \tag{31}$$

and $\delta_{t,t'} = 1$ iff. $t = t'$. Let us assume that $\{x_t\}_{t\geq 0}$ is (wide sense) stationary and purely non-deterministic. This together with an observability condition on (F, H) implies that $\{h_t\}_{t\geq 0}$ is (wide-sense) stationary and purely non-deterministic as well. Let $P = \mathbb{E}[h_t h_t^T]$; P satisfies

$$P = FPF^T + Q, \tag{32}$$

which in turn implies that F has all its eigenvalues in the open unit disc. Finally the covariance function of x_t is given by

$$r_0 = \mathbb{E}[x_t^2] = R + HPH^T; \tag{33}$$

$$\text{for all } k \in \mathbb{N}^*, r_k = \mathbb{E}[x_t x_{t-k}] = HF^{k-1}\underbrace{(FPH^T + S)}_{N}. \tag{34}$$

Starting from a covariance $(r_k)_{k\in\mathbb{N}}$, the SR problem consists in building a minimal "Markovian representation" of $(x_t)_{t\in\mathbb{N}}$, i.e. a state-space system (30)-(31), with F of minimal dimension.

Step 1: Thanks to the structure of function $(r_k)_{k\in\mathbb{N}^*}$, we can as in section IV-A build a Hankel matrix

$$\mathcal{R}_\infty = \begin{bmatrix} r_1 & r_2 & r_3 & \dots \\ r_2 & r_3 & & \\ r_3 & & & \\ \vdots & & & \end{bmatrix} = \begin{bmatrix} H \\ HF \\ HF^2 \\ \vdots \end{bmatrix} \begin{bmatrix} N & FN & F^2N & \dots \end{bmatrix} \tag{35}$$

which should be compared to factorization (29). The first (and, in fact, "deterministic") step of a SR algorithm consists in building a minimal realization (H, F, N) of $(r_k)_{k\in\mathbb{N}^*}$ (unique up to an invertible matrix);

Step 2: At this point, we dispose of (H, F, N) but N remains a function of P and S (see (34)), and it remains to identify Q and R. This second step is more delicate for the problem must be solved under positivity constraints: P and $\begin{bmatrix} Q & S \\ S^T & R \end{bmatrix}$ are covariance matrices and so must be semi-definite positive (≥ 0). If these contraints were not statisfied, the solution would be meaningless. Finally, the problem is as follows: knowing (H, F, N, r_0), we look for (P, Q, R, S) such that

$$\begin{bmatrix} P & N \\ N^T & r_0 \end{bmatrix} - \begin{bmatrix} F \\ H \end{bmatrix} P \begin{bmatrix} F^T & H^T \end{bmatrix} = \begin{bmatrix} Q & S \\ S^T & R \end{bmatrix}, \tag{36}$$

$$P > 0, \tag{37}$$

$$\begin{bmatrix} Q & S \\ S^T & R \end{bmatrix} \geq 0, \tag{38}$$

in which > 0 stands for definite positive (the constraint on P should be, *a priori*, that P is *semi-definite* positive, but indeed it happens that any solution P must be *definite* positive [39] (see theorem 1 below), whence (37)). Let us notice that equation (36) gives the covariance of (h_{t+1}, x_t). Since (u_t, v_t) is a white noise, this covariance satisfies a (Ricatti) equation of the same kind as that satisfied by P (equation (32), which in fact is a submatrix of (36)).

System (36) can be seen as a system with three equations and four unknowns (P, Q, R and S), or rather as a system with three equations and three unknowns (Q, R and S), parameterized by P. Finally, P parameterizes solutions of the constrained system (36)-(38). Let $\mathcal{P}$ be the set of parameters

$$\mathcal{P} = \{P \text{ s.t. } (36) - (38) \text{ are satisfied}\}. \tag{39}$$

Positive real lemma, positivity of $(r_k)_{k\in\mathbb{N}}$, structure of $\mathcal{P}$

A result known as the *positive real lemma* (initially proved in the spectral domain) connects the positivity of the series $(r_k)_{k\in\mathbb{N}}$ (in other words, whether $(r_k)_{k\in\mathbb{N}}$ is a *covariance* series) to the existence of at least one solution to the constrained system (36)-(38). Let us recall that the infinite series $(r_k)_{k\in\mathbb{N}}$ is a covariance series iff. the Toeplitz form $\sum_{i,j=0}^{m} u_i u_j r_{|j-i|}$ is positive or null for all m, i.e. iff. the associated Toeplitz matrix is semi-definite positive for all m.

Lemma 1 (Positive real lemma [39]): The series $(r_k)_{k\in\mathbb{N}}$ is a covariance series iff. $\mathcal{P}$ is non void.

We now consider the structure of $\mathcal{P}$.

Theorem 1 ([39]): The set $\mathcal{P}$ is closed, convex, bounded and definite positive; it admits (for the usual order relation between symmetric matrices) a maximum P^* and a minimum P_*.

Let us finally notice that there exist efficient algorithms for building elements of $\mathcal{P}$ (see [39], [41]).

V. Expressivity of GUM, HMC and RNN

We are now ready to come back to mapping (25) (23) (24), and first need to study the range of ϕ.

A. Algebraic properties induced by the factorizability and positivity constraints

The range of ϕ is strictly included into $\mathbb{R}^{\mathbb{N}}$, since $(r_k)_{k\in\mathbb{N}}$ in (25) is indeed a *factorized* and *covariance* series. As we now see, it is possible to characterize this range, via algebraic tests which determine whether a given real series satisfies these two constraints.

1) Factorizability: First, factorization (24) implies that the doubly infinite Hankel matrix built on $(r_k)_{k\in\mathbb{N}^*}$ factorizes as (35). So the rank of $\mathcal{R}_\infty$ is finite and lower than or equal to n (the dimension of F), and is equal to n if and only if each factor has itself full rank n. In this case, (H, F, N) is a so-called minimal (deterministic) realization of $(r_k)_{k\in\mathbb{N}^*}$ (see section IV for more details). One can show (see [45, proposition 3] or section IV) that all minimal realizations are isomorphic: (H_1, F_1, N_1) and (H_2, F_2, N_2) are two minimal realizations of $(r_k)_{k\in\mathbb{N}^*}$ if and only if there exists $T_{1,2}$ invertible such that

$$(H_2, F_2, N_2) = (H_1T_{12}^{-1}, T_{12}F_1T_{12}^{-1}, T_{12}N_1). \tag{40}$$

2) Positivity: Apart from being factorizable, any sequence $(r_k)_{k\in\mathbb{N}} \overset{(25)}{=} \phi(\theta)$ is also a covariance series, which can be characterized either by the constraint that for all $k \in \mathbb{N}$, the Toeplitz matrix with first row $[r_0, \cdots, r_k]$ is positive semi definite or, equivalently, that $C(z) \overset{def}{=} r_0 + 2\sum_{k=1}^{\infty} r_k z^k$ is a Carathéodory function, ie. has positive real part in the open unit disk $\{z \in \mathbb{C}; |z| < 1\}$ (Carathéodory-Toeplitz theorem, see e.g. [48]).

In the context of this paper, it is however more interesting to recall the positive real lemma, which relies on the factorizability constraint we just evoked. So assume that rank$(\mathcal{R}_\infty)$ is finite, which enables to build a minimal set (F, H, N) satisfying $r_k = HF^{k-1}N$ for all k, $k \geq 1$. As we recalled in section IV, positivity of the series $(r_k)_{k\in\mathbb{N}}$ is related to whether there exists at least one matrix $P > 0$ satisfying

$$\begin{bmatrix} P & N \\ N^T & r_0 \end{bmatrix} - \begin{bmatrix} F \\ H \end{bmatrix} P \begin{bmatrix} F^T & H^T \end{bmatrix} \geq 0, \tag{41}$$

three unknowns (Q, R and S) parameterized by P. i.e. whether the set $\mathcal{P}$ defined in (39) is non void.

B. Compared expressivity of the three models

Let us summarize section V-A.

- Starting from any real valued series $(r_k)_{k\in\mathbb{N}}$, this series is factorizable (i.e., there exists a triplet (H, F, N) such that $r_k = HF^{k-1}N$ for all $k \in \mathbb{N}^*$) iff. the Hankel matrix $\mathcal{R}_\infty$ is finite rank; the rank n of $\mathcal{R}_\infty$ is also the minimal dimension of any realization of $(r_k)_{k\in\mathbb{N}}$;
- Starting from a factorizable series $(r_k)_{k\in\mathbb{N}}$, this series is a covariance series if and only if there exists at least one matrix $P > 0$ satisfying (41).

This discussion is summarized in Fig. 2 below, the South-West part of which is the range of function ϕ in (25).

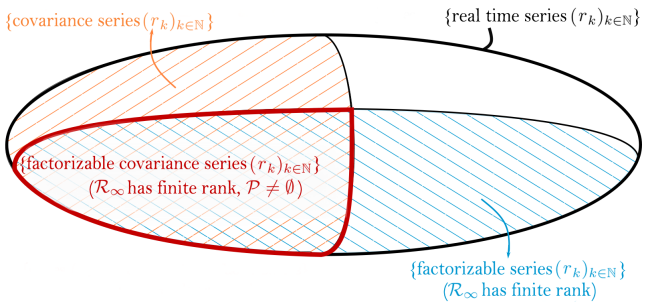


Fig. 2: This figure represents the set of all real times series. The series $(r_k)_{k\in\mathbb{N}}$ which are factorizable covariance series is the South-West quarter of the figure (orange and blue lines). Computing $\mathcal{R}_\infty$ enables to move from the full set to the Southern part, whereas the positive real lemma enables to move from the Southern part to the South-West quarter.

We now study if any point of the South-West corner (i.e., any factorizable covariance function) can be realized by a GUM, an HMC and/or an RNN.

1) Expressivity of GUM: This question does not raise any particular difficulty. Since $(r_k)_{k\in\mathbb{N}}$ is a factorized covariance function, it can be realized (see section IV) by the state space system (30)-(31) for some (F, H, Q, R, S). System (30) can be rewritten (if $R \neq 0$) as

$$\begin{cases} h_{t+1} &= ah_t + cx_t + u'_t \\ x_t &= bh_t + v' - t \end{cases}, \tag{42}$$

$$\mathbb{E}\left[\begin{bmatrix} u'_t \\ v'_t \end{bmatrix} \cdot \begin{bmatrix} u_t'^T & v'_t \end{bmatrix}\right] = \begin{bmatrix} \alpha & 0 \\ 0 & \beta \end{bmatrix}, \tag{43}$$

in which

$$a = F - SR^{-1}H, b = H, c = SR^{-1}, \tag{44}$$

$$\begin{bmatrix} \alpha & 0 \\ 0 & \beta \end{bmatrix} = \begin{bmatrix} I & -SR^{-1} \\ 0 & 1 \end{bmatrix} \underbrace{\begin{bmatrix} Q & S \\ S^T & R \end{bmatrix}}_{\geq 0} \begin{bmatrix} I & 0 \\ -R^{-1}S^T & 1 \end{bmatrix}; \tag{45}$$

Eq. (45) ensures that $\begin{bmatrix} \alpha & 0 \\ 0 & \beta \end{bmatrix} \geq 0$ (and thus $\alpha \geq 0$). Equations (42)-(43) are a state space representation of (11) (15)-(17). In other words, any point of the South-West corner can be realized by some linear and Gaussian GUM model.

2) Expressivity of HMC: We know that any factorizable covariance function $(r_k)_{k\in\mathbb{N}}$ such that $\dim(\mathcal{R}_\infty) = n$ can be realized by a GUM of dimension n. Starting from such a series, under which conditions does there exist an HMC of the same degree n which produces that same covariance series? We have the following result (see Appendix B for a proof).

Proposition 2: Let $(r_k)_{k\in\mathbb{N}}$ a factorizable covariance function and let (H, F, N) a triplet (with F of minimal dimension n) produced by DR. The series $(r_k)_{k\in\mathbb{N}}$ can be realized by an HMC of dimension n if and only if there exists $\tilde{P}$ (and thus $\tilde{Q}(\tilde{P})$ et $\tilde{R}(\tilde{P})$) such that

$$\begin{bmatrix} \tilde{P} & N \\ N^T & r_0 \end{bmatrix} - \begin{bmatrix} F \\ H \end{bmatrix} \tilde{P} \begin{bmatrix} F^T & H^T \end{bmatrix} = \begin{bmatrix} \tilde{Q} & 0 \\ 0 & \tilde{R} \end{bmatrix}, \tag{46}$$

$$\tilde{P} > 0, \tag{47}$$

$$\begin{bmatrix} \tilde{Q} & 0 \\ 0 & \tilde{R} \end{bmatrix} \geq 0. \tag{48}$$

Remark 2: Finally, let $\tilde{\mathcal{P}}$ the set of solutions $\tilde{P}$ of the constrained problem (46)-(48). One can note that $\tilde{\mathcal{P}}$ is a convex subset of $\mathcal{P}$. On the other hand, as compared to $\mathcal{P}$, (46) yields the supplementary constraint (74). This equation can be satisfied only if $N \in \text{Span}(F)$. Moreover, if F is invertible, (74) also implies

$$H^T F^{-1} N > 0. \tag{49}$$

So if (74) and/or (49) is not satisfied, then the series $(r_k)_{k\in\mathbb{N}}$ cannot be realized by an HMC of dimension n.

3) Expressivity of D-GUM and RNN: Similarly as the HMC, the study can be done from any triplet (H, F, N) provided by the DR step. We have to take into account the D-GUM constraint $\alpha = 0$ (or $Q - SR^{-1}S^T = 0$, see (44)-(45)) and the RNN constraint $\eta = c\text{Var}(x_0)c^T$ (or $P = SR^{-1}r_0R^{-T}S^T$).

Proposition 3: Let $(r_k)_{k\in\mathbb{N}}$ a factorizable covariance function and let (H, F, N) a triplet (with F of minimal dimension

n) produced by DR. Let us note $\mathcal{P}$ the set of solutions of system (36)-(38). Then $(r_k)_{k\in\mathbb{N}}$ can be realized by a D-GUM if and only if there exists $\tilde{P} \in \mathcal{P}$ such that

$$\tilde{P}-F\tilde{P}F^T-(N-F\tilde{P}H^T)(r_0-H\tilde{P}H^T)^{-1}(N-F\tilde{P}H^T)^T=0. \quad (50)$$

If in addition $\tilde{P}$ satisfies

$$\tilde{P} = r_0(r_0-H\tilde{P}H^T)^{-2}(N-F\tilde{P}H^T)(N-F\tilde{P}H^T)^T, \quad (51)$$

the covariance series can be produced by a traditional RNN initialized to $h_0 = 0$, with a linear activation function.

Remark 3: Note that by construction $\tilde{P}$ in (51) is a rank 1 $n \times n$ semi-definitive positive matrix, and is positive definite only if $n = 1$. In other words, a factorizable covariance series can be realized by an RNN if the latent vector is monodimensional and (51) holds (as we will check in the worked example below), but can never be realized by an RNN if $n > 1$.

C. A worked example (unidimensional case)

We now illustrate the preceding section in the scalar case. We first look for conditions on the triplet (H, F, N) such that a factorizable series $(r_k)_{k\in\mathbb{N}}$ is a covariance function. We next give conditions on this triplet to determine if the covariance series can be produced by one of the generative models of this paper.

1) SR step: Let $(r_k)_{k\in\mathbb{N}}$ be a covariance series, factorizable as $HF^{k-1}N$ for all $k \in \mathbb{N}^*$ with H, F and N scalar. We assume that such a triplet (H, F, N) has been produced by DR of $(r_k)_{k\in\mathbb{N}}$, and we search for the scalar parameters P, Q, R, S satisfying (36)-(38). Each of these equations becomes respectively

$$\begin{cases} Q &= P(1-F^2) \\ R &= r_0 - PH^2 \\ S &= N - HFP \end{cases}, \quad (52)$$

$$P \geq 0, \quad (53)$$

$$\begin{cases} QR - S^2 &\geq 0 \\ Q &\geq 0 \\ R &\geq 0 \end{cases}. \quad (54)$$

In particular, (54) corresponds to the semi-definite positive constraint (38) when Q, R and S are scalar.

Let us build the set $\mathcal{P}$ of positive numbers P which satisfy this system. The second inequality of (54) is satisfied when $F^2 \leq 1$; by using (52), the first inequality of (54) reads

$$\Xi_{/H^2}(P) \stackrel{\text{def.}}{=} -H^2P^2+[r_0(1-F^2)+2HFN]P-N^2 \geq 0. \quad (55)$$

Since polynomial $\Xi_{/H^2}$ is concave, one can show easily that (55) admits a solution provided

$$\frac{r_0(F-1)}{2} \leq HN \leq \frac{r_0(F+1)}{2}, \quad (56)$$

and that $\mathcal{P}$ is included in $[P_{/H^2,1}, P_{/H^2,2}]$ with

$$P_{/H^2,i} = \frac{(2HFN + r_0(1-F^2)) + (-1)^i\sqrt{\delta}}{2H^2}, \quad (57)$$

$$\delta = (1-F^2)(r_0(1+F)-2HN)(r_0(1-F)+2HN). \quad (58)$$

Moreover, constraints (53) and $R \geq 0$ in (54) imply that P belongs to $[0, \frac{r_0}{H^2}]$; but it can be checked that $[P_{/H^2,1}, P_{/H^2,2}] \subseteq [0, \frac{r_0}{H^2}]$ so constraints $P \geq 0$ and $R \geq 0$ do not yield further interval restrictions. Finally, factorizable series $(r_k)_{k\in\mathbb{N}}$ is a covariance function if $F^2 \leq 1$ and HN satisfies (56); $\mathcal{P}$ then coincides with

$$\mathcal{P} = [P_{/H^2,1}, P_{/H^2,2}] \quad (59)$$

and is non void. It can also be produced by a GUM whose parameters are deduced from (44)-(45).

Remark 4: Finally one can show easily that $P_{/H^2,1} > 0$ if $N > 0$ (the case $P_{/H^2,1} = 0$ is possible only if $\Xi_{/H^2}(0) = -N^2 \geq 0$, and so $N = 0$, which corresponds to the degenerate case of a series $(r_k)_{k\in\mathbb{N}}$ which is null everywhere except at $k = 0$ where it is equal to r_0), so $\mathcal{P}$ is a *definite* positive set, which is in concordance with Faurre's theory [39, theorem 7].

2) HMC case: As a consequence of Proposition 2, a factorizable covariance series $(r_k)_{k\in\mathbb{N}}$ can be produced by an HMC if there exists $\tilde{P}$ in $\mathcal{P}$ such that $\tilde{P} = N(HF)^{-1}$; equivalently, $\tilde{P}$ has to satisfy $\tilde{P} = N(HF)^{-1}$ and $\tilde{P} \in (0, r_0H^{-2}]$. So condition (56) for HN becomes

$$\begin{cases} 0 < HN \leq r_0F, & \text{if } F \geq 0 \\ r_0F \leq HN < 0, & \text{if } F \leq 0 \end{cases}.$$

3) D-GUM and RNN cases: Remember that for a D-GUM, the first inequality of (54) becomes an equality. So polynomial $\Xi_{/H^2}$ in (55) is equal to zero, and system (52)-(54) admits two solutions, $P_{/H^2,1}$ and $P_{/H^2,2}$. Consequently, as the GUM, a D-GUM can produce any covariance series, but requires less parameters since $\alpha = 0$.

Finally, the additional RNN constraint becomes $P = r_0S^2R^{-2}$. In the same time P has to satisfy $P = P_{/H^2,1}$ or $P = P_{/H^2,2}$. Using elementary calculus, these new systems have a solution if $HN = r_0F$, or $HN = r_0F(2F^2-1)$.

A graphical representation of the expressivity of each generative model in function of parameters F and HN is given in Fig. 3 below (these 1-dimensional results coincide with those obtained in [49] by using the Caratheodory theorem).

VI. Conclusion

In this paper we adressed a comparative study of HMC and RNN, which are familiar tools for predicting time series. Even though both tools were developed in different communities, we first showed that they indeed share close features when the RNN is turned into a generative model, and thus when HMC and RNN are considered as two latent variables probabilistic models with close enough (conditional) independence structures. Under this framework, both structures can be seen as two different particular instances of a common generative unified model. We next compared both models from the point of view of *expressivity*, i.e. the relative complexity of the joint probability distribution of an observations sequence, induced by the underlying latent variables. By contrast with previous studies, which were of an experimental nature, our approach consisted in thoroughly quantifying the modeling power of both models. To that end we considered the linear and Gaussian assumption, which induces that the probability

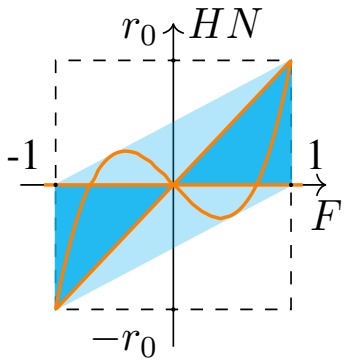


Fig. 3: Expressivity of RNN, HMC, D-GUM and GUM with regards to parameters F and HN in the scalar case. The parallelogram (blue and cyan) coincides with the factorizable covariance series $r_k = F^{k-1}HN$. Such series can be produced by a GUM or a D-GUM. The blue (resp. orange) area (resp. curves) coincides with the value of F and HN which can be taken by the HMC (resp. the RNN).

distributions of an observations sequence produced by each model are characterized by structured covariance series, which enabled us to call for SRT. Finally we provided implicit conditions under which a given covariance series (and thus a given probability distribution) can be realized by a GUM, an HMC and/or a D-GUM or an RNN. These implicit conditions turn to an explicit cartography of the models in the mono-dimensional case.

APPENDIX

A. Inference algorithms in HMC

1) Computing the likelihood: As we recalled in section II-A, being able to compute and maximize the likelihood is a key factor for choosing a probabilistic model. The likelihood can be computed from the predictive likelihoods $p_\theta(x_s|x_{0:s-1})$. In model (8), for all s, $0 \leq s \leq t$, we have

$$p_\theta(x_s|x_{0:s-1}) = \int \underbrace{p_\theta(h_{s-1}|x_{0:s-1})}_{\text{filtering pdf}} \underbrace{p_\theta(h_s|h_{s-1})p_\theta(x_s|h_s)}_{\text{HMC transition pdf}} \mathrm{d}h_{s-1:s}, \quad (60)$$

where $p_\theta(x_s|h_s)$ and $p_\theta(h_s|h_{s-1})$ are the elementary factors in (8). On the other hand, the filtering pdf $p_\theta(h_{s-1}|x_{0:s-1})$ can be computed recursively:

$$p_\theta(h_s|x_{0:s}) = \frac{p_\theta(x_s|h_s)}{p_\theta(x_s|x_{0:s-1})} \times \int p_\theta(h_s|h_{s-1})p_\theta(h_{s-1}|x_{0:s-1})\mathrm{d}h_{s-1}. \quad (61)$$

So equations (60) and (61) enable to compute the predictive pdf $p_\theta(x_s|x_{0:s-1})$ and the filtering pdf $p_\theta(h_s|x_{0:s})$ recursively. Given the initial pdf $p_\theta(h_0)$, we first compute $p_\theta(x_s|x_{0:s-1})$ from $p_\theta(h_{s-1}|x_{0:s-1})$ via (60). We next compute $p_\theta(h_s|x_{0:s})$ from $p_\theta(h_{s-1}|x_{0:s-1})$, $p_\theta(x_s|x_{0:s-1})$ and the HMC transition pdfs. Note that it is the HMC structure (8) that enables this likelihood calculation, at least theoretically (in practice, the integrals in equations (60) and (61) can be difficult to compute, see section A3 for further discussion).

2) Learning: In latent variables models (as is the case here) computing the maximum likelihood estimate is difficult, and one generally resorts to approximations. In particular, the EM algorithm is an iterative learning method which runs as follows. At step i, we first compute, under parameter θ_i, the expected log-likelihood given observations $x_{0:t}$:

$$\mathbb{E}_{\theta_i}\left[\log p_\theta(x_{0:t}, h_{0:t})|x_{0:t}\right] = \int p_{\theta_i}(h_{0:t}|x_{0:t}) \log p_\theta(x_{0:t}, h_{0:t})\mathrm{d}h_{0:t} \quad (62)$$

$$= \sum_{s=1}^{t} \int \left[\log p_\theta(x_s|h_s) + \log p_\theta(h_s|h_{s-1})\right] p_\theta(h_{s-1:s}|x_{0:t})\mathrm{d}h_{s-1:s} + \int \log\left(p_\theta(x_0|h_0)p_\theta(h_0)\right)\mathrm{d}h_0, \quad (63)$$

next we update this parameter $\theta_i \to \theta_{i+1}$ by maximizing

$$\theta_{i+1} = \underset{\theta}{\text{argmax}}\ \mathbb{E}_{\theta_i}\left[\log p_\theta(h_{0:t}, x_{0:t})|x_{0:t}\right]. \quad (64)$$

Equations (62) and (64) are respectively the E and M steps of the EM algorithm. In particular, the E step is feasible if factors $p_{\theta_i}(h_{s-1}, h_s|x_{0:t})$ and $p_{\theta_i}(h_0|x_{0:t})$ can be computed. As is well known, the algorithm ensures that the likelihood increases with the iterations: for all i, $p_{\theta_i}(x_{0:t}) \leq p_{\theta_{i+1}}(x_{0:t})$. Stronger theoretical guarantees are available under further conditions [50][51].

3) Practical considerations: In practice, computing the likelihood and maximizing it via the EM algorithm depend on the model assumptions. We can distinguish three different cases.

Case 1: Linear and Gaussian state space systems.

Assume that $x_{0:t}$ and $h_{0:t}$ take continuous values, and that the transition pdfs $p_\theta(h_{s+1}|h_s)$ and $p_\theta(x_s|h_s)$ are linear and Gaussian: $h_{s+1} = F_s h_s + u_s$, $x_s = G_s h_s + v_s$ where F_s and G_s are matrices and h_0 and (u_s, v_s) are Gaussian independent random vectors. Under such assumptions all pdfs of interest are indeed Gaussian, so propagating them through time reduces to propagating their parameters.

Computing the likelihood in this model can be done via an iterative algorithm known as the Kalman filter (KF), introduced in the control community in the 1960's [52], [53], [54] and heavily studied since then [55], [56], [57]. The KF enables to compute efficiently the filtering pdf $p_\theta(h_s|x_{0:s})$ for any s. Similarly, one can show that the parameters of $p_\theta(h_{s-1}, h_s|x_{0:t})$ (see (63)) and of the smoothing pdf $p_\theta(h_s|x_{0:t})$ (which are also Gaussian) can be computed via backward propagation [58], which enables an efficient implementation of the EM algorithm.

Case 2: continuous states (general case).

In the general case (non linear transition pdfs and/or non Gaussian noise), exact computing is not available and one needs to resort to approximations. Approximation methods include the extended KF, i.e. a KF in a linearized model [59]

[55], and the unscented KF, which propagates an approximation of the one- and second-order moments of the pdfs of interest [60] [61] [62].

Particle filtering (or sequential Monte Carlo) methods are another class of approximate solutions [63] [64] [65], which consist in propagating a random, discrete approximation of $p_\theta(h_{0:t}|x_{0:t})$, via an importance sampling mecanism with resampling [66]. Let us start from $\hat{p}_\theta(h_{0:t}|x_{0:t}) = \sum_{i=1}^N w_t^{(i)} \delta_{h_{0:t}^{(i)}}(h_{0:t})$, where δ is the Dirac mass and $w_t^{(i)}$ are a normalized set of weights. The weighted trajectories $\{h_{0:t}^{(i)}, w_t^{(i)}\}_{i=1}^N$ are propagated via three steps. For all $s \in \mathbb{N}$, the i^{th} particle is sampled from a conditional importance pdf q:

$$\tilde{h}_{s+1}^{(i)} \sim q(h_{s+1}|h_s^{(i)}). \tag{65}$$

Next we compute its unnormalized weight

$$\tilde{w}_{s+1}^{u,(i)} = w_s^{(i)} p_\theta\left(\tilde{h}_{s+1}^{(i)}|h_s^{(i)}\right) p_\theta\left(x_{s+1}|\tilde{h}_{s+1}^{(i)}\right) / q\left(\tilde{h}_{s+1}^{(i)}|h_s^{(i)}\right), \tag{66}$$

which is normalized as $\tilde{w}_{s+1}^{(i)} = \tilde{w}_{s+1}^{u,(i)} / \sum_{j=1}^N \tilde{w}_{s+1}^{u,(j)}$. Finally the trajectories can be resampled, i.e. $h_{0:s+1}^{(i)} \sim \sum_{j=1}^N \tilde{w}_{s+1}^{(j)} \delta_{(h_{0:s}^{(j)}, \tilde{h}_{s+1}^{(j)})}(h_{0:s+1})$, and given new weights $w_{s+1}^{(i)} = \frac{1}{N}$. This optional resampling step keeps a larger proportion of trajectories with strong weights, to the detriment of those of low weight. If the trajectories are not resampled, then $\tilde{h}_{s+1}^{(i)}$ (resp. $\tilde{w}_{s+1}^{(i)}$) reduces to $h_{s+1}^{(i)}$ (resp. $w_{s+1}^{(i)}$). With or without resampling, the procedure is repeated from (65).

The unnormalized weights computed in (66) enable in turn to compute an approximation of the predictive likelihood: $\hat{p}_\theta(x_t|x_{0:t-1}) = \frac{1}{N}\sum_{i=1}^N \tilde{w}_t^{u,(i)}$, from which an estimate of the likelihood is computed from (7). Finally $\hat{p}_\theta(h_{0:t}|x_{0:t})$ also provides an approximation of (62):

$$\hat{\mathbb{E}}_{\theta'}\left[\log p_\theta(h_{0:t}, x_{0:t})|x_{0:t}\right] = \sum_{i=1}^N w_{\theta',t}^{(i)} \log \hat{p}_\theta(h_{0:t}^{(i)}, x_{0:t}),$$

which still remains to be maximized (notation $w_{\theta',t}^{(i)}$ recalls that weights are built from parameter θ', see (66)). In practice approximation $\hat{p}$ can be poor, in particular when N is very small w.r.t. t. As a possible rescue one can use particle smoothing algorithms [67] [68] [69] [70], which aim at improving the approximation of $p(h_{s-1}, h_s|x_{0:t})$ in (63).

Case 3 : discrete latent states.

HMC with discrete latent states were introduced in the 1960's [71] [72] [73] and have been used in such fields as langage processing [19] [20], bioinformatics [74] or digital communications [75] [73].

In the discrete case, the problem is that computing the likelihood as $p_\theta(x_{0:t}) = \sum_{h_{0:t}} p_\theta(h_{0:t}, x_{0:t})$, i.e. via brute force marginalization of the full joint pdf, is unfeasible due to the exponential cost. The success of HMC comes from the fact that the computation of the likelihood $p_\theta(x_{0:t})$ can be performed in linear time. Indeed the likelihood can be seen as another marginalized pdf:

$$p_\theta(x_{0:t}) = \sum_{h_t} \underbrace{p_\theta(h_s, x_{0:s})}_{\alpha(h_s)}, \tag{67}$$

in which pdfs $\alpha(h_s)$ can be computed recursively in linear cost in the *forward* time direction:

$$\alpha(h_0) = p_\theta(h_0) p_\theta(x_0|h_0) \tag{68}$$

$$\alpha(h_{s+1}) = p_\theta(x_{s+1}|h_{s+1}) \sum_{h_s} p_\theta(h_{s+1}|h_s)\alpha(h_s). \tag{69}$$

Note that $\alpha(h_s)$ is proportional to the filtering probability mass function, and that (69) is the discrete analog of (61). As for the predictive likelihood, it reads

$$p(x_{s+1}|x_{0:s}) = \frac{\sum_{h_{s+1}} p_\theta(x_{s+1}|h_{s+1}) \sum_{h_s} p(h_{s+1}|h_s)\alpha(h_s)}{\sum_{h_s}\alpha(h_s)}.$$

From (63) (where integrals become sums), running the EM algorithm requires calculating, for all s, $0 \le s \le t$, pdf $p_\theta(h_{s-1}, h_s|x_{0:t})$, which is proportional to

$$p_\theta(h_{s-1}, h_s, x_{0:t}) = \underbrace{p_\theta(x_{s+1:t}|h_s)}_{\beta(h_s)} \alpha(h_{s-1}) p_\theta(h_s|h_{s-1}) p_\theta(x_s|h_s). \tag{70}$$

In particular, pdf $p_\theta(h_{s-1}, h_s, x_{0:t})$ depends on the backward pdfs $\beta(h_s) = p_\theta(x_{s+1:t}|h_s)$ which, similarly to pdfs $\alpha(h_s)$, can be computed recursively at linear cost, but in the reverse time direction (whence the term *backward*):

$$\beta(h_t) = 1 \tag{71}$$

$$\beta(h_s) = \sum_{h_{s+1}} \beta(h_{s+1}) p_{\theta_k}(x_{s+1}|h_{s+1}) p_\theta(h_{s+1}|h_s). \tag{72}$$

The recursive calculation of fonctions $\alpha(h_s)$ and $\beta(h_s)$, for all s, is the so called forward-backward algorithm [71] [72] [20]. Finally from (70) we have

$$p_\theta(h_{s-1}, h_s|x_{0:t}) = \frac{\beta(h_s)\alpha(h_{s-1}) p_\theta(h_s|h_{s-1}) p_\theta(x_s|h_s)}{\sum_{h_{s-1},h_s} \beta(h_s)\alpha(h_{s-1}) p_\theta(h_s|h_{s-1}) p_\theta(x_s|h_s)}. \tag{73}$$

It remains to maximize w.r.t. $p_\theta(h_t|h_{t-1})$ and $p_\theta(x_t|h_t)$. Computing $p_\theta(h_{s-1}, h_s, x_{0:t})$ enables to update these pmfs / pdfs in the M step of the EM algorithm; this version of the EM algorithm, applied to discrete latent states HMC, is called the Baum-Welch algorithm [76], [20]. Finally observe that the forward-backward algorithm enables to compute the smoothing pmf (for a given, fixed parameter θ), since from (73) we get $p_\theta(h_s|x_{0:t}) = \frac{\alpha(h_s)\beta(h_s)}{\sum_{h_s}\alpha(h_s)\beta(h_s)}$.

B. Proof of Proposition 2

According to Remark 1, the problem reduces to studying whether among the set of all solutions, there exists at least one such that $c = 0$, and thus (see (44)) $S = 0$. So we need to solve (41) with the additional constraint

$$N = FPH^T, \tag{74}$$

whence (46). However, this raises the following question: although the set $\mathcal{P}_1$ in which we look for an HMC solution is built from *a* triplet (H_1, F_1, N_1), produced by the DR step, this triplet is not unique; if $\mathcal{P}_1$ had no HMC solution, could another set $\mathcal{P}_2$, built from another triplet (H_2, F_2, N_2), nevertheless contain an HMC solution?

We thus need to study the relation between $\mathcal{P}_1$ and $\mathcal{P}_2$. Since (H_1, F_1, N_1) is of minimal degree, from (40) any other minimal degree solution (H_2, F_2, N_2) can be computed from (H_1, F_1, N_1) via an invertible matrix T_{12}. Let P_1 be an element of $\mathcal{P}_1$. By pre- (respectively post-) multiplying equation (41), with parameters (H_1, F_1, N_1), by $\begin{bmatrix} T_{12} & 0 \\ 0 & 1 \end{bmatrix}$ (respectively $\begin{bmatrix} T_{12} & 0 \\ 0 & 1 \end{bmatrix}^T$), one can show easily that

$$\mathcal{P}_2 = \{T_{12} P_1 T_{12}^T \; ; \; P_1 \in \mathcal{P}_1\}, \tag{75}$$

which we denote simply by the set equation $\mathcal{P}_2 = T_{12}\mathcal{P}_1 T_{12}^T$. This is summarized by Figure 4 below.

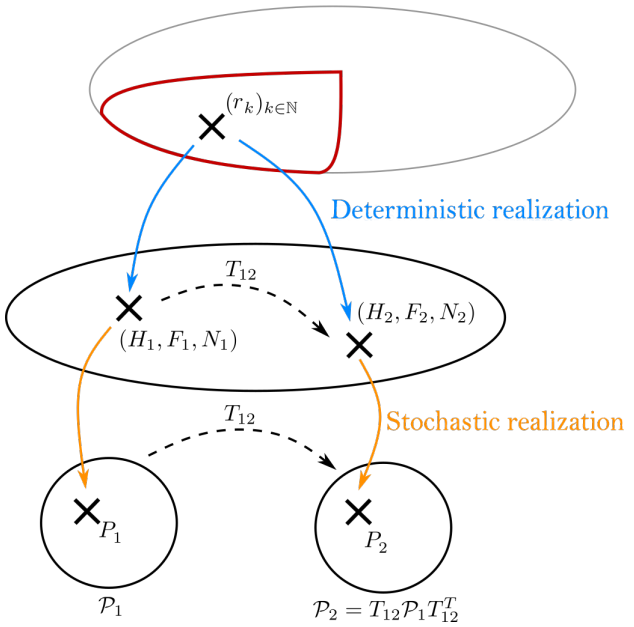


Fig. 4: Starting from a factorizable covariance series $(r_k)_{k\in\mathbb{N}}$ (see figure 2), the deterministic realization step (in blue) consists in finding a triplet (H, F, N) representing the series under study. This step provides one solution out of an infinity of solutions to this problem. These solutions are isomorphic and it suffices to know the appropriate invertible matrix T_{12} to move from a given solution (H_1, F_1, N_1) to another (H_2, F_2, N_2) (dashed arrow).
The SR step (in orange) amounts to finding a state-space system modeling function $(r_k)_{k\in\mathbb{N}}$ from the triplet (H, F, N) obtained at the previous step. A triplet (H_i, F_i, N_i) leads to a set of solutions $\mathcal{P}_i$. These sets are also isomorphic and T_{12} suffices for moving from $\mathcal{P}_1$ to $\mathcal{P}_2$, respectively obtained from (H_1, F_1, N_1) and (H_2, F_2, N_2) (dashed arrow).

Let now a triplet (H_1, F_1, N_1) and a solution $P_1 \in \mathcal{P}_1$, such that

$$N_1 = F_1 P_1 H_1^T. \tag{76}$$

This matrix P_1 is thus an HMC solution of $(r_k)_{k\in\mathbb{N}}$. Equation (76) is equivalent to

$$\underbrace{T_{12}N_1}_{N_2} = \underbrace{T_{12}F_1T_{12}^{-1}}_{F_2}\underbrace{T_{12}P_1T_{12}^T}_{P_2\in\mathcal{P}_2}\underbrace{T_{12}^{-T}H_1^T}_{H_2^T}; \tag{77}$$

so from (40) and (75), we see that $P_2 = T_{12}P_1T_{12}^T$ is one HMC element belonging to set $\mathcal{P}_2$. In other words, if there exists an HMC element in the set $\mathcal{P}_1$ associated to a triplet (H_1, F_1, N_1) from the equivalence class produced by the DR step, then any set $\mathcal{P}_2 = T_{12}\mathcal{P}_1T_{12}^T$ (with T_{12} an arbitrary invertible matrix) also contains an HMC solution. Similarly, if there is no such element in $\mathcal{P}_1$, then no set $\mathcal{P}_2 = T_{12}\mathcal{P}_1T_{12}^T$ will contain a solution either. Finally it suffices to look for an HMC solution in the set $\mathcal{P}$ produced by the SR algorithm, without bothering any longer of the other elements in the equivalence class.

REFERENCES

[1] R. H. Shumway and D. S. Stoffer, *Time Series Analysis and Its Applications*. Springer, 2000.
[2] R. Douc, E. Moulines, and D. Stoffer, *Nonlinear time series: Theory, methods and applications with R examples*. CRC press, 2014.
[3] A. M. Deshmukh, "Comparison of hidden markov model and recurrent neural network in automatic speech recognition," *European Journal of Engineering and Technology Research*, vol. 5, pp. 958–965, August 2020.
[4] R. Bikmukhamedov, A. Nadeev, G. Maione, and D. Striccoli, "Comparison of HMM and RNN models for network traffic modeling," *Internet Technology Letters*, vol. 3, 01 2020.
[5] A. Borovkov, *Statistique mathématique*. Editions Mir, 1987.
[6] L. Wasserman, *All of statistics: a concise course in statistical inference*. Springer Science & Business Media, 2013.
[7] A. P. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society (B)*, vol. 39, no. 1, pp. 1–38, 1977.
[8] C. M. Bishop, *Pattern recognition and machine learning*. Springer, 2006.
[9] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, 2009.
[10] J. H. Manton and P.-O. Amblard, "A primer on Reproducing Kernel Hilbert Spaces," *Found. Trends Signal Process.*, vol. 8, no. 1–2, p. 1–126, 2015.
[11] V. I. Paulsen and M. Raghupathi, *An introduction to the theory of reproducing kernel Hilbert spaces*, vol. 152. Cambridge University Press, 2016.
[12] A. K. Jain, J. Mao, and K. M. Mohiuddin, "Artificial neural networks: A tutorial," *Computer*, vol. 29, no. 3, pp. 31–44, 1996.
[13] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
[14] C. J. Burges, "A tutorial on support vector machines for pattern recognition," *Data mining and knowledge discovery*, vol. 2, no. 2, pp. 121–167, 1998.
[15] W. Hu, Y. Liao, and V. R. Vemuri, "Robust anomaly detection using support vector machines," in *Proceedings of the international conference on machine learning*, pp. 282–289, Citeseer, 2003.
[16] V. Vapnik, *The nature of statistical learning theory*. Springer science & business media, 2013.
[17] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*, vol. 1. MIT press Cambridge, 2016.
[18] V. N. Vapnik, *Statistical Learning Theory*. Wiley-Interscience, 1998.
[19] L. R. Rabiner and B. H. Juang, "An introduction to hidden Markov models," *IEEE ASSP Magazine*, pp. 4–16, January 1986.
[20] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[21] J. T. Connor, R. D. Martin, and L. E. Atlas, "Recurrent neural networks and robust time series prediction," *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 240–254, 1994.

[22] O. Cappé, É. Moulines, and T. Rydén, *Inference in Hidden Markov Models*. Springer Series in Statistics, Springer-Verlag, 2005.

[23] T. Mikolov, S. Kombrink, L. Burget, J. Černockỳ, and S. Khudanpur, "Extensions of recurrent neural network language model," in *2011 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 5528–5531, IEEE, 2011.

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[25] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of control, signals and systems*, vol. 2, no. 4, pp. 303–314, 1989.

[26] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural networks*, vol. 4, no. 2, pp. 251–257, 1991.

[27] A. Pinkus, "Approximation theory of the MLP model in neural networks," *Acta numerica*, vol. 8, no. 1, pp. 143–195, 1999.

[28] Z. Lu, H. Pu, F. Wang, Z. Hu, and L. Wang, "The expressive power of neural networks: A view from the width," in *Advances in neural information processing systems*, pp. 6231–6239, 2017.

[29] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," tech. rep., California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[30] A. Robinson and F. Fallside, *The utility driven dynamic error propagation network*. University of Cambridge Department of Engineering Cambridge, MA, 1987.

[31] P. J. Werbos, "Backpropagation through time: what it does and how to do it," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.

[32] M. C. Mozer, "A focused backpropagation algorithm for temporal," *Backpropagation: Theory, architectures, and applications*, vol. 137, 1995.

[33] Y. Goldberg, "Neural network methods for natural language processing," *Synthesis Lectures on Human Language Technologies*, vol. 10, no. 1, pp. 1–309, 2017.

[34] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *International conference on machine learning*, pp. 1310–1318, 2013.

[35] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *NIPS 2014 Workshop on Deep Learning*, 2014.

[36] M. Gevers and W. Wouters, "An innovations approach to the discrete-time stochastic realization problem," *Journal A*, vol. 19, no. 2, pp. 90–110, 1978.

[37] R. W. Brockett, *Finite dimensional linear systems*. SIAM, 2015.

[38] P. L. Faurre, "Stochastic realization algorithms," in *Mathematics in Science and Engineering*, vol. 126, pp. 1–25, Elsevier, 1976.

[39] P. Faurre, *Opérateurs rationnels positifs*. Dunod, 1979.

[40] M. Gevers, "A personal view of the development of system identification: A 30-year journey through an exciting field," *IEEE Control systems magazine*, vol. 26, no. 6, pp. 93–105, 2006.

[41] P. E. Caines, *Linear stochastic systems*, vol. 77. SIAM, 2018.

[42] C.-T. Chen, *Introduction to linear system theory*. Holt, Rinehart and Winston, 1970.

[43] T. Kailath, *Linear systems*, vol. 156. Prentice-Hall Englewood Cliffs, NJ, 1980.

[44] C. K. Chui and G. Chen, *Signal processing and systems theory: selected topics*, vol. 26. Springer Science & Business Media, 2012.

[45] B. Ho and R. E. Kalman, "Effective construction of linear state-variable models from input/output functions," *at-Automatisierungstechnik*, vol. 14, no. 1-12, pp. 545–548, 1966.

[46] L. S. de Jong, *Numerical aspects of realization algorithms in linear systems theory*. PhD thesis, Department of Mathematics and Computer Science, 1975.

[47] L. S. de Jong, "Numerical aspects of recursive realization algorithms," *SIAM Journal on Control and optimization*, vol. 16, no. 4, pp. 646–659, 1978.

[48] N. I. Akhiezer and N. Kemmer, *The classical moment problem and some related questions in analysis*, vol. 5. Oliver & Boyd Edinburgh, 1965.

[49] A. Salaün, Y. Petetin, and F. Desbouvries, "Comparing the modeling powers of RNN and HMM," in *18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, pp. 1496–1499, IEEE, 2019.

[50] C. J. Wu, "On the convergence properties of the EM algorithm," *The Annals of statistics*, pp. 95–103, 1983.

[51] S. Balakrishnan, M. J. Wainwright, B. Yu, *et al.*, "Statistical guarantees for the EM algorithm: From population to sample-based analysis," *Annals of Statistics*, vol. 45, no. 1, pp. 77–120, 2017.

[52] R. E. Kalman, "A new approach to linear filtering and prediction problems," *J. Basic Eng., Trans. ASME, Series D*, vol. 82, no. 1, pp. 35–45, 1960.

[53] R. E. Kalman and R. S. Bucy, "New results in linear filtering and prediction theory," *J. Basic Eng., Trans. ASME, Series D*, vol. 83, no. 3, pp. 95–108, 1961.

[54] Y. Ho and R. Lee, "A Bayesian approach to problems in stochastic estimation and control," *IEEE transactions on automatic control*, vol. 9, no. 4, pp. 333–339, 1964.

[55] B. D. Anderson and J. B. Moore, *Optimal filtering*. Courier Corporation, 2012.

[56] T. Kailath, A. H. Sayed, and B. Hassibi, *Linear estimation*. Prentice Hall Information and System Sciences Series, Upper Saddle River, New Jersey: Prentice Hall, 2000.

[57] R. J. Meinhold and N. D. Singpurwalla, "Understanding the Kalman filter," *The American Statistician*, vol. 37, no. 2, pp. 123–127, 1983.

[58] R. H. Shumway and D. Stoffer, "An approach to time series smoothing and forecasting using the EM algorithm," *Journal of Time Series Analysis*, vol. 3, no. 4, pp. 253–264, 1982.

[59] G. Chen, *Approximate Kalman filtering*, vol. 2. World Scientific, 1993.

[60] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, 2004.

[61] S. Julier, J. Uhlmann, and H. F. Durrant-Whyte, "A new method for the nonlinear transformation of means and covariances in filters and estimators," *IEEE Transactions on automatic control*, vol. 45, no. 3, pp. 477–482, 2000.

[62] H. M. Menegaz, J. Y. Ishihara, G. A. Borges, and A. N. Vargas, "A systematization of the unscented Kalman filter theory," *IEEE Transactions on automatic control*, vol. 60, no. 10, pp. 2583–2598, 2015.

[63] A. Doucet, N. de Freitas, and N. Gordon, "Sequential Monte Carlo methods in practice," *Information Science and Statistics (Springer New York, 2001)*, 2001.

[64] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Transactions on signal processing*, vol. 50, no. 2, pp. 174–188, 2002.

[65] N. Chopin and O. Papaspiliopoulos, *An introduction to sequential Monte Carlo*. Springer, 2020.

[66] H. Kahn and A. W. Marshall, "Methods of reducing sample size in Monte Carlo computations," *Journal of the Operations Research Society of America*, vol. 1, no. 5, pp. 263–278, 1953.

[67] N. Kantas, A. Doucet, S. S. Singh, J. Maciejowski, and N. Chopin, "On particle methods for parameter estimation in state-space models," *Statist. Sci.*, vol. 30, pp. 328–351, August 2015.

[68] M. Briers, A. Doucet, and S. Maskell, "Smoothing algorithms for state–space models," *Annals of the Institute of Statistical Mathematics*, vol. 62, no. 1, p. 61, 2010.

[69] C. M. Carvalho, M. S. Johannes, H. F. Lopes, N. G. Polson, *et al.*, "Particle learning and smoothing," *Statistical Science*, vol. 25, no. 1, pp. 88–106, 2010.

[70] P. Fearnhead, D. Wyncoll, and J. Tawn, "A sequential smoothing algorithm with linear computational cost," *Biometrika*, vol. 97, no. 2, pp. 447–464, 2010.

[71] L. E. Baum and T. Petrie, "Statistical inference for probabilistic functions of finite state Markov chains," *The annals of mathematical statistics*, vol. 37, no. 6, pp. 1554–1563, 1966.

[72] L. E. Baum and J. A. Eagon, "An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology," *Bulletin of the American Mathematical Society*, vol. 73, no. 3, pp. 360–363, 1967.

[73] G. D. Forney, "The Viterbi algorithm," *Proceedings of the IEEE*, vol. 61, pp. 268–78, March 1973.

[74] T. Koski, *Hidden Markov models for bioinformatics*, vol. 2. Springer Science & Business Media, 2001.

[75] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimal decoding algorithm," *IEEE Transactions on Information Theory*, vol. 13, pp. 260–69, April 1967.

[76] L. E. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains," *The annals of mathematical statistics*, vol. 41, no. 1, pp. 164–171, 1970.