Predictive Linear-Gaussian Models of Stochastic Dynamical Systems with Vector-Valued Actions and Observations*

Matthew Rudary and **Satinder Singh**
{mrudary,baveja}@umich.edu
Computer Science and Engineering, University of Michigan, Ann Arbor, MI 48105 USA

Abstract

Predictive state representations use probabilities of future events as the state of a partially observable system, as opposed to most classical models, which use probabilistic statements about latent variables as state. We present a new version of the predictive linear-Gaussian model (PLG), a predictive state representation that models discrete-time dynamical systems with real-*vector*-valued actions and observations. This extends earlier work on PLGs in which the dynamical systems were limited to *scalar* observations. We show that the new PLG subsumes linear dynamical systems (LDSs, sometimes called Kalman filter models) of equal dimension. Finally, we introduce an algorithm to estimate PLG parameters from data and show that our algorithm is a consistent estimation procedure.

1 Introduction

Linear dynamical system models (LDSs), also known as state-space models and Kalman filter models, are an important class of models of dynamical systems. They are used to control and make predictions about dynamical systems in a wide variety of applications, and are very useful when their parameters are known or can be estimated with the help of domain knowledge. However, their parameters are not easily learned from data. The predictive linear-Gaussian model (PLG), a predictive state representation for discrete-time stochastic dynamical systems with continuous *scalar* observations, was introduced by Rudary et al. (2005; 2006). They showed that PLGs were equivalent to LDSs with scalar observations. Predictive state representations (PSRs) like the PLG use predictions about future events in the system as their state representation, rather than (for example) probability distributions over latent variables. PSRs have advantages over some traditional models. For example, in domains with discrete observations, PSRs have more expressive power than hidden Markov models and partially observable Markov decision processes (POMDPs) (Jaeger 1997; Singh, James, & Rudary 2004). In addition, Wolfe, James, & Singh (2005) showed that learning one type of PSR from data outperforms learning POMDPs using expectation maximization on several simulated problems.

Outside of the PLG, work on PSRs has focused mainly on modeling dynamical systems with discrete observations (Jaeger 1997; Littman, Sutton, & Singh 2001), though see Section 4 for related work. Here, we extend the PLG model to domains with *vector*-valued observations and show that the new PLG subsumes the LDS model. This is a critical advance, as nearly all problems in robotics, control, and other fields require that multiple observations be modeled. Additionally, we present a parameter estimation algorithm for PLGs that is non-iterative and provably consistent.

2 The PLG Model

Like all models of dynamical systems, the PLG computes probability distributions over future events given the history of past interactions with the system. It does this by summarizing the information contained in this interaction into a compact representation called the model's *state*; the semantics of the state and how it is updated in the face of new information is where models differ. This section will describe how the PLG uses the mean and variance of *future* observations as state.

We begin with some notation. In the systems modeled by the PLG, the world is observed at discrete points in time, starting with $t = 1$. At each time step, an action (or control) vector u_t is provided as input to the system. This is a real-valued vector of length l, but it is not modeled as a random variable; in particular, the model does not require that actions take on any particular (distribution of) values.[1] An observation (or measurement) Y_t is then observed. This is a real-valued vector of length m, modeled as a random variable. The sequence of actions and observations from time 1 up to and including t is denoted h_t, the history of interaction with the system up to time t.

2.1 The Scalar PLG

To aid the reader in understanding the vector-valued PLG, we first present the scalar PLG originally described by

*The authors were supported by NSF grant IIS-0413004. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

[1] Though we shall see that the parameter estimation algorithm does have such requirements to guarantee consistency.

Rudary, Singh, & Wingate (2005). The scalar PLG was designed to model dynamical systems with no actions ($l = 0$) and scalar observations ($m = 1$). The state of the scalar PLG is the mean and variance of the next n observations; n is called the *dimension* of the system. That is, the information given by the history h_t is summed up in the distribution of *future* observations.

More precisely, let Z_t be a random vector that collects the n observations following t; that is,

$$Z_t = \begin{pmatrix} Y_{t+1} \\ Y_{t+2} \\ \vdots \\ Y_{t+n} \end{pmatrix}.$$

The state of the scalar PLG at time t (i.e. just after observing Y_t) is the mean (μ_t) and variance (Σ_t) of $Z_t|h_t$. This can be written as $Z_t|h_t \sim \mathcal{N}(\mu_t, \Sigma_t)$. The fact that the state of the model is given by predictions of the future (in the form of a probability distribution over future observations) is the reason that the PLG is called a predictive state model.

Given these state semantics, it still remains to define the parameters of the PLG and how those parameters are used to update the state. This is begun by defining the core dynamics of the model:

$$Z_{t+1} = GZ_t + \eta_{t+1},$$

where η_{t+1} is a noise term that captures the stochasticity of the model. What this means is that Z_t evolves according to a noisy linear function G. But G is heavily constrained because of the semantics of Z_t. For instance, the first element of Z_{t+1} and the second element of Z_t are both equal to Y_{t+2}. In fact, $Z_t^i = Z_{t+1}^{i-1}$ for $i = 2, 3, \ldots, n$ (it will be useful hereafter to refer compactly to specific elements of Z_t and other vectors; the ith element of Z_t is denoted by Z_t^i). The only "new" element of Z_{t+1} is its last, $Z_{t+1}^n = Y_{t+n+1}$. This means that only the nth row of G is made up of free parameters of the PLG; the remaining elements are the same for all scalar PLGs.

Up to this point, the scalar PLG may seem quite similar to the autoregressive (AR) model (Pandit & Wu 1983), which uses the values of the *previous* n observations as state and also has linear dynamics. The key difference is in how the noise term is defined. In the AR models (as well as many others), the noise term at each time step would be an i.i.d. Gaussian variable. In the scalar PLG, η_{t+1} is still a Gaussian variable, but it is allowed to *covary* with Z_t; this gives the PLG representational power equivalent to the LDS (which allows infinite memory and properly subsumes the AR model).

To be more concrete, we fix the first $n - 1$ elements of η_{t+1} to be 0. This is because it is necessary to apply additive noise to a particular observation only once; as noted above, a given observation Y_{t+n+1} appears as an element of $Z_t, Z_{t+1}, \cdots, Z_{t+n-1}$. When multiple noise terms apply to the same observation, it leads to conflicting probability distributions depending on which noise term is being considered. By fixing the first $n - 1$ elements to zero, the noise associated with Y_{t+n+1} will be introduced at time t into Z_t, and will be carried into future iterations by the linear function G. So we now define the distribution of η_{t+1}^n given a history h_t as follows:

$$\eta_{t+1}^n | h_t \sim \mathcal{N}(0, \sigma) \quad \text{and} \quad \mathrm{Cov}[Z_t, \eta_{t+1}^n | h_t] = C.$$

Here σ is the scalar and time-independent variance of η_{t+1}, and C is the vector-valued and time-independent covariance of Z_t with η_{t+1}.

Together with the initial state μ_0 and Σ_0, G, σ, and C are the parameters of the scalar PLG. Using the dynamics given above, a pair of state update equations can be derived that compute the next state (μ_{t+1} and Σ_{t+1}) from the previous state (μ_t and Σ_t) whenever a new observation $Y_{t+1} = y_{t+1}$ is seen.[2] The derivation is as follows. Z_{t+1} and Y_{t+1} are jointly distributed Gaussian random variables, which can be seen by the following relation:

$$\begin{pmatrix} \\ Z_t \\ \\ Y_{t+n+1} \end{pmatrix} = \begin{pmatrix} Y_t \\ Y_{t+1} \\ \vdots \\ Y_{t+n+1} \end{pmatrix} = \begin{pmatrix} Y_t \\ \\ Z_{t+1} \\ \\ \end{pmatrix}.$$

The distribution of $Z_t|h_t$ is known (it is the state at time t), and the distribution of $Y_{t+n+1}|h_t$ (as well as $\mathrm{Cov}[Z_t, Y_{t+n+1}|h_t]$) can be computed from the equation $Y_{t+n+1} = G_{n,\cdot} Z_t + \eta_{t+1}^n$, where $G_{n,\cdot}$ is the nth row of G. These distributions can be used to compute the distribution of $Z_{t+1}|(h_t, Y_{t+1} = y_{t+1})$, whose mean and variance are the next state. (We omit the details of the update equations here; they can be taken as a special case of Equations (4) and (5) developed in this paper for the more general vector-valued case.)

Again, the covariance C is the key to the power of this model; whereas an AR model's "memory" only goes n time steps into the past (because its state is the *value* of the last n observations), the fact that the observations covary with one another means that an observation's distribution can be affected by an event arbitrarily far into the past, giving it significant representational ability. In particular, the n-dimensional scalar PLG can model any system that an n-dimensional linear dynamical system[3] can (this was demonstrated by Rudary, Singh, & Wingate (2005) and is also a special case of our Theorem 1).

2.2 The Vector-Valued PLG

When extending the scalar PLG to deal with vector-valued observations and actions, there are several challenges to overcome. The most basic, and probably the most important, is to decide which observations comprise Z_t and then work out the details of the dynamics based on this decision. The effects of actions must also be determined. We must also show that the vector-valued PLG can represent any system that an LDS of the same dimension can; as we shall see in Section 2.3, this will require a slight change to the next-observation semantics of the model presented here.

[2] This introduces a convention used hereafter: The realization of a random variable is set in lowercase, while the variable itself is set in uppercase.

[3] The LDS model is discussed in more detail in Section 2.4.

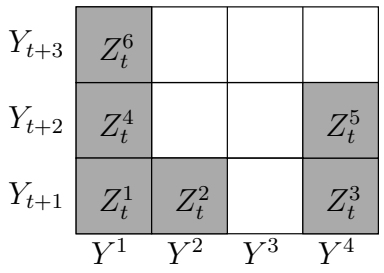

Figure 1: This figure shows the makeup of Z_t in a system with $n = 6$ and $m = 4$. This figure forms a skyline—no box is used in Z_t unless the boxes below it are also used.

As with the scalar version, the state of the vector-valued PLG is the mean (μ_t) and variance (Σ_t) of $Z_t|h_t$, where Z_t is an n-dimensional vector of future observations; n is called the *dimension* of the PLG model. As before, $Z_t|h_t$ is distributed according to a multivariate Gaussian distribution. The PLG then uses this distribution to compute the distribution of any desired future observation(s). By comparison, in an n-dimensional LDS model, the distribution of future observations is computed from the distribution of X_t, an n-dimensional latent state vector.

Makeup of Z_t The concept at the heart of the vector-valued PLG is how Z_t is constructed. Each element of the vector Z_t is an element of an observation vector that will be observed after time t. In the scalar PLG, Z_t is just the concatenation of the next n observations $Y_{t+1}, Y_{t+2}, \ldots, Y_{t+n}$. It would be possible to use the concatenation of the next n observations in the vector-valued PLG as well; in the class of systems we consider (n-dimensional LDSs), knowledge of the distribution of this vector would be sufficient to compute the distribution of any future observation(s). However, this is inefficient—this Z_t would contain mn elements when just n are sufficient. The solution, then, is to pick a subset of those mn elements. We select n *observational elements*—that is, elements of observation vectors—whose joint distribution is sufficient to reconstruct the distribution of all future observations. It is of course not generally the case that all such subsets are sufficient; the particular subset chosen depends on the system to be modeled.

For a number of reasons, there are some subsets of the next mn observations that will not be permitted. In particular, if the jth element of the future observation Y_{t+k} is included as part of Z_t, then the jth elements of $Y_{t+1}, Y_{t+2}, \ldots, Y_{t+k-1}$ must be included as well. We call this the "skyline" requirement, because when we make a diagram as in Figure 1, it forms a silhouette of a city skyline. This restriction does not reduce the representative power of the PLG.

One reason for the skyline requirement is the consistency of the noise terms. In the scalar PLG, only the *last* term of η_{t+1} was non-zero because we wanted only one noise variable associated with each observation. If the skyline requirement were violated, an observation would be in non-consecutive Z vectors. Considering the Z_t layout shown in Figure 1, suppose that the fourth element of Z_t were removed. Then Y^1_{t+3} would be an element of Z_t and Z_{t+2}, but not Z_{t+1}. It is not tracked at $t + 1$; when it is reintroduced at $t + 2$, should another noise term be applied? This would break the assumption that each observational element has a single noise term associated with it and would lead to conflicting definitions of its distribution. On the other hand, the fact that it is not tracked at $t + 1$ means that its distributional information is essentially lost, and so information about the original noise term is discarded and must be re-introduced, which suggests that another noise term *should* be used. This conflict argues in favor of the skyline requirement.

Another reason to introduce the skyline requirement is efficiency in the parameters. When an observational element is a member of consecutive Z vectors, one row of the linear trend G is predetermined to copy that element. The PLG referred to by Figure 1, for example, has a G whose fourth row copies the sixth element (Y^1_{t+3}) onto the fourth ($Y^1_{\mathbf{t+2}}$) when t increases. When an observational element is in non-consecutive Z vectors, this copying would not be possible, and so an additional row must be made up of free parameters.

The elements that make up Z_t must be selected so that the distribution of $Z_t|h_t$ is sufficient to compute the distribution of any future observation(s) conditioned on the history h_t. We also require that they be selected so that the next observation is a linear function of Z_t; i.e., $Y_{t+1} = JZ_t$, where the $m \times n$ matrix J is a parameter of the model. In many cases when $m \leq n$, this can be satisfied by including every element of Y_{t+1} as the first m elements of Z_t and setting J to be the first m rows of the $n \times n$ identity matrix. While this is a limiting assumption, Section 2.3 will provide a more general solution.

Core Dynamics of the PLG The core dynamics of the model are linear; in an uncontrolled system,

$$Z_{t+1} = GZ_t + \eta_{t+1}, \quad t = 1, 2, \ldots, \tag{1}$$

where G is the linear trend in the observations (a parameter of the model) and η_{t+1} is a noise term. The distribution of this noise term is a choice that greatly affects the dynamics of the model. If η_{t+1} is chosen to be i.i.d. Gaussian noise, as in classical models like the LDS or the autoregressive (AR) model, this model would be quite similar to a vector-valued version of the AR model, which has considerably less representational power than the LDS. However, in our model, we allow η_{t+1} to *covary* with Z_t; as we will show, this gives the PLG power equivalent to that of the LDS. In particular, given a history h_t,

$$\eta_{t+1}|h_t \sim \mathcal{N}(\mathbf{0}, \Sigma_\eta) \quad \text{and} \quad \mathrm{Cov}[Z_t, \eta_{t+1}|h_t] = C_\eta.$$

There is a restriction on Σ_η and C_η. It is possible that a particular observational element Y^i_t may appear in Z_s for multiple values of s. For example, in Figure 1, the observational element Y^1_{t+3} appears as Z^6_t, Z^4_{t+1}, and Z^1_{t+2}. In general, this *will* happen whenever $n > m$ and may also happen when $n \leq m$. As in the case of the scalar PLG, we want

only a single noise term to apply to each observational element to prevent conflicting probability distributions. Therefore, we fix $\eta^i_{t+1} = 0$ when Z^i_t is a later appearance of an observational element than the first. Equivalently, η^i_{t+1} has 0 variance and 0 covariance with Z_t. Thus, in the example, the first and fourth elements of η_t will be fixed at 0 for all t, but the sixth element may be a nontrivial random variable, because Z^6_t is the earliest appearance of Y^1_{t+3} in a Z vector. The nonzero elements of η_{t+1} correspond to the top shaded box of each "stack" in Figure 1.

Modeling Actions We follow the approach of Rudary & Singh (2006) in modeling actions. That is, u_t, the action at time t, affects the expected value (but not the variance) of future observations—that is, $Y_{t+1}, Y_{t+2}, \ldots$. However, the actions are *not* modeled as random variables. Formally, each action's effect on the means of future observations is a linear function of the action, which leads to the following dynamical equation:

$$Z_{t+1} = GZ_t + \sum_{i=1}^{\tau_{\max}} \Gamma_i u_{t+i} + \eta_{t+1}. \tag{2}$$

The model parameters $\Gamma_1, \ldots, \Gamma_{\tau_{\max}}$ are $n \times l$ matrices that describe the linear effects of the actions, and are parameters of the PLG. The upper limit of the sum, $\tau_{\max}$, is the maximum look-ahead horizon of Z_t. That is, Z_t contains at least one element of $Y_{t+\tau_{\max}}$ but no elements of $Y_{t+\tau_{\max}+i}, i > 0$ (in the example of Figure 1, $\tau_{\max} = 3$). Therefore, Z_{t+1} contains at least one element of $Y_{t+\tau_{\max}+1}$, which can be affected by the action $u_{t+\tau_{\max}}$ but not by $u_{t+\tau_{\max}+1}$; hence, $\tau_{\max}$ is the upper limit on the summation.

As with the noise terms, in order for the effects of the actions to be well-defined, they should only apply to the *first* appearance of a particular observational element in a Z_t vector. This is accomplished by fixing the ith row of Γ_k to be $\mathbf{0}$ when Z^i_t is not the first appearance of an observational element. Thus, as with the noise term, in the running example, the first and fourth rows of Γ_1, Γ_2, and Γ_3 are fixed at $\mathbf{0}$, but the sixth rows of each of these parameters may take on any value. In addition, causality must be preserved: no action may have an effect on elements of Z_{t+1} that will be observed *before* that action will be taken. In the example, this means that the second row of Γ_1 may be nonzero, but the second rows of Γ_2 and Γ_3 must be $\mathbf{0}$, because the actions u_{t+2} and u_{t+3} cannot affect elements of Y_{t+2}.

In controlled systems, the distribution of $Z_t|h_t$ is undefined without information about the actions that will be executed between now (t) and the time that the elements of Z_t will be observed. Because the actions are not modeled as random variables, it is impossible to marginalize out their effects. This is important, because the PLG's state is defined in terms of the distribution of Z_t. The simplest solution is to assume that future actions will be $\mathbf{0}$; because of the actions' linear effect on the future, this is the assumption that actions that have not yet been taken will not affect the values of future observations.

But note that this is just an assumption to make the state well-defined; the dynamics do not require that the actions be zero in order for the model to be correct. When the actual value of an action is determined, its effects are propagated onto the state using the state update equations (which we define below).

State Update As described at the beginning of this section, the state of the PLG is the mean and variance of Z_t, conditioned on the history of interaction (and assuming future actions to be $\mathbf{0}$):

$$Z_t|h_t, u_{t+1} = \mathbf{0}, u_{t+2} = \mathbf{0}, \ldots \sim \mathcal{N}(\mu_t, \Sigma_t). \tag{3}$$

With each new time step, a new observation becomes available and another action has been taken. We must update the state of the system to take this information into account. In order to do this, we must calculate the distribution of Z_{t+1} conditioned on both h_t and the new observation $Y_{t+1} = y_{t+1}$. This is a relatively straightforward computation, as $Y_{t+1}|h_t$ and $Z_{t+1}|h_t$ are jointly Gaussian random variables; the conditional distribution of $Z_{t+1}|h_t, Y_{t+1} = y_{t+1}$ is readily calculated using a standard result (e.g. Catlin 1989). Using this result, we obtain the updated values:

$$\mu_{t+1} = G\mu_t + Lu_t + f_t(J\Sigma_t J^\top)^{-1}(y_{t+1} - J\mu_t) \tag{4}$$

$$\Sigma_{t+1} = G\Sigma_t G^\top + GC_\eta + C_\eta^\top G^\top + \Sigma_\eta - f_t(J\Sigma_t J^\top)^{-1} f_t^\top, \tag{5}$$

where $f_t = (G\Sigma_t + C_\eta^\top)J^\top$ is the covariance of Y_{t+1} and Z_{t+1} given h_t. L describes the cumulative effect that the action has on all the observations of Z_{t+1}:

$$Lu = \mathrm{E}[Z_{t+1}|h_t, u_{t+1} = u, u_{t+2} = \mathbf{0}, \ldots] - \mathrm{E}[Z_{t+1}|h_t, u_{t+1} = \mathbf{0}, u_{t+2} = \mathbf{0}, \ldots].$$

This is computed from $\Gamma_1, \ldots, \Gamma_{\tau_{\max}}$ by

$$L \triangleq \Gamma_1 + G\Gamma_2 + G^2\Gamma_3 + \cdots + G^{\tau_{\max}-1}\Gamma_{\tau_{\max}}.$$

We can now define the basic PLG model, though recall that a limiting assumption will be resolved in the next section.

Definition 1. *The Predictive Linear-Gaussian model is defined by a) the state semantics of* (3)*, b) the state update of* (4) *and* (5)*, and c) the parameters: the linear trend, G; the linear effects of the action, $\Gamma_1, \ldots, \Gamma_{\tau_{\max}}$, the "next-observation function," J; the noise parameters, Σ_η and C_η; and the initial state, μ_0 and Σ_0.*

The PLG can be used to compute the probability density of any trajectory $s_T = y_1, y_2, \ldots, y_T$ as follows: Initialize the state $\mu \leftarrow \mu_0$ and $\Sigma \leftarrow \Sigma_0$ and the density $p(s_T) \leftarrow 1$. Then, for $t = 1, \ldots, T$, set $p(s_T) \leftarrow p(s_T) \cdot p_{\mathcal{N}}(y_t; J\mu, J\Sigma J^\top)$, where $p_{\mathcal{N}}(x; \nu, V)$ is the density at x of a multivariate Gaussian distribution with mean ν and variance V; update μ and Σ according to (4) and (5).

2.3 The Variance-Adjusted PLG

The next-observations dynamics of the PLG, $Y_{t+1} = JZ_t$, places some limitations on the systems that PLGs can model.

For instance, it precludes most models in which the observation vector has a larger dimension than the model itself (that is, $m > n$). More generally, the PLG is unable to model systems in which the observation vector space is underranked in expectation—that is, when $\{\mathrm{E}[Y_{t+1}|h_t] : h_t \text{ a length-}t \text{ history}\}$ has dimension less than m.[4]

The limitation arises because the variance need not be underranked even when the means *are* linearly dependent—yet this relationship is forced by $Y_{t+1} = JZ_t$. When that equation holds, $\mathrm{Var}[Y_{t+1}|h_t] = J\Sigma_t J^\top$, which will be underranked whenever J is.

For example, suppose that we wish to control an oven in an industrial process. The oven has two temperature sensors that are unbiased but whose readings differ from the actual temperature inside the oven by i.i.d. Gaussian noise;[5] these sensors are the only observations (i.e. $m = 2$). In this case, $\mathrm{E}[Y^1_{t+1}|h_t] = \mathrm{E}[Y^2_{t+1}|h_t]$, so the mean vector has a trivial linear dependency. However, because the noise experienced by the two sensors is *independent*, it is *not* the case that $Y^1_{t+1}|h_t = Y^2_{t+1}|h_t$. The linear dependence in the means does not carry over to the variance of the two elements of the observation vector. Thus, there is no J that satisfies $Y_{t+1} = JZ_t$.

To eliminate this problem, we introduce a variant of the PLG: the *variance-adjusted PLG*. It is identical to the basic PLG except the next observation is distributed according to

$$Y_{t+1}|h_t \sim \mathcal{N}(J\mu_t, J\Sigma_t J^\top + \Sigma_{\mathrm{adj}}),$$

where Σ_{adj} is a symmetric matrix. It need not be a valid covariance matrix (i.e. symmetric positive semidefinite), but the sum $J\Sigma_t J^\top + \Sigma_{\mathrm{adj}}$ must be symmetric positive semidefinite for all t. This new formulation requires that the state update equations be altered slightly:

$$\mu_{t+1} = G\mu_t + f_t(J\Sigma_t J^\top + \Sigma_{\mathrm{adj}})^{-1}(y_{t+1} - J\mu_t) \quad (6)$$

$$\Sigma_{t+1} = G\Sigma_t G^\top + GC_\eta + C_\eta^\top G^\top + \Sigma_\eta - f_t(J\Sigma_t J^\top + \Sigma_{\mathrm{adj}})^{-1} f_t^\top. \quad (7)$$

Note that the standard PLG is a special case of the variance-adjusted PLG in which $\Sigma_{\mathrm{adj}} = \mathbf{0}$.

2.4 Equivalence of PLG and LDS

Now that the PLG and its variance-adjusted variant have been defined, it is natural to ask if they model anything useful. In fact, an n-dimensional variance-adjusted PLG can model any system that can be modeled by an n-dimensional LDS. Before stating this result as a theorem and sketching its proof, we review the LDS model.

The LDS Model The LDS model is based on a pair of stochastic processes. The first is the latent (unobservable) X_t process, which evolves through a linear function with i.i.d. Gaussian noise. The second is the observation process Y_t, which is a linear function of the first, again with i.i.d. Gaussian noise. X_t is an n-vector and Y_t is an m-vector. The dynamics are governed by the following:

$$\begin{aligned} X_1 &\sim \mathcal{N}(x_1^-, P_1^-), & \mathrm{E}[\omega_t] = \mathrm{E}[\nu_t] = \mathrm{E}[\omega_t \nu_s^\top] = \mathbf{0}, \\ X_{t+1} &= AX_t + \omega_{t+1}, & \mathrm{Cov}[\omega_t, \omega_s] = \delta_{t,s} Q, \\ Y_t &= HX_t + \nu_t, & \mathrm{Cov}[\nu_t, \nu_s] = \delta_{t,s} R, \end{aligned}$$

where $\delta_{i,j}$, the Kronecker delta, is 1 if $i = j$ and 0 otherwise. The state of the LDS can be tracked with the Kalman filter (Kalman 1960; Welch & Bishop 2004), which maintains the state variables $x_t^- = \mathrm{E}[X_t|h_{t-1}]$ and $P_t^- = \mathrm{Var}[X_t|h_{t-1}]$ through the following equations:

$$\begin{aligned} K_t &= P_t^- H^\top (HP_t^- H^\top + R)^{-1}, \\ x_t^+ &= x_t^- + K_t(y_t - Hx_t^-), & P_t^+ &= P_t^- - K_t H P_t^-, \\ x_{t+1}^- &= Ax_t^+ + Bu_t, & P_{t+1}^- &= AP_t^+ A^\top + Q. \end{aligned}$$

K_t is known as the *Kalman gain* at time t, and x_t^+ and P_t^+ have the semantics $X_t|h_t \sim \mathcal{N}(x_t^+, P_t^+)$.

Equivalence Theorem We now state our main result.

Theorem 1. *Every LDS with n-dimensional state has an equivalent n-dimensional variance-adjusted PLG.*

If the LDS parameter H has rank m, it can be shown that $\Sigma_{\mathrm{adj}} = \mathbf{0}$, and thus its equivalent PLG is a standard (non-variance-adjusted) PLG.

The full proof of Theorem 1 is similar in outline to the proofs of similar theorems for scalar PLGs (Rudary, Singh, & Wingate 2005; Rudary & Singh 2006). However, several details differ, and the proof can be found in an online appendix (Rudary & Singh 2008). However, we will point out a few salient features of that proof here.

The strategy is to construct an equivalent n-dimensional PLG given an n-dimensional LDS. This is done by finding a mapping from the Kalman filter's state variables x_{t+1}^- and P_{t+1}^- to the PLG's state variables μ_t and Σ_t such that both models compute the same distribution for $Y_{t+1}|h_t$, then computing PLG parameters that preserve this property through the state update. An inductive argument then shows that both models compute the same distribution over trajectories of arbitrary length, showing equivalence of the models. This is the same strategy as in the scalar case.

One of the major differences in the vector-valued case is in the first step: finding a mapping from Kalman filter state to PLG state. In order to do that, we must populate Z_t. In the scalar case, Z_t was simply the next n observations; here, Z_t is populated by taking the first n spanning rows of $\mathcal{M}_n$, which is defined by

$$\mathcal{M}_n = \begin{pmatrix} H \\ HA \\ \vdots \\ HA^{n-1} \end{pmatrix}; \text{ note: } \mathcal{M}_n x_{t+1}^- = \begin{pmatrix} E[Y_{t+1}|h_t] \\ E[Y_{t+2}|h_t] \\ \vdots \\ E[Y_{t+n}|h_t] \end{pmatrix}.$$

Thus, each row of $\mathcal{M}_n$ corresponds to an element of a future observation. The first n rows of $\mathcal{M}_n$ that span the matrix are combined to form the matrix M; the observation

[4] Except in certain special cases when the observation's variance matrix has a particular set of linear dependencies.

[5] While each sensor may differ from the actual temperature by an i.i.d. Gaussian random variable, the process as a whole may still have non-i.i.d. noise.

element corresponding to the ith row of M is chosen as Z_t^i. We say "the first n rows" that span $\mathcal{M}_n$ because $\mathcal{M}_n$ may have rank less than n, though it will not have rank more than n. It is most efficient to choose n so that $\mathrm{rk}(\mathcal{M}_n) = n$.

We now define the state mapping. μ_t can be computed directly: $\mu_t = \mathrm{E}[Z_t|h_t] = Mx_{t+1}^-$. A more involved derivation allows us to compute the covariance matrix Σ_t. Z_t can be written in terms of the LDS random variables as $Z_t = MX_t + E_t$, where each element of E_t is a linear combination of (elements of) the LDS noise terms ε_{t+i} and ν_{t+j}—as defined in Section 2.4—for various i and j. Thus, $\Sigma_t = \mathrm{Cov}[Z_t, Z_t|h_t] = MP_t^-M^\top + \Phi$, where $\Phi = \mathrm{Cov}[E_t, E_t|h_t]$ is independent of time and its (i,j)th element can be written as

$$\Phi_{ij} = H_{\rho(i),\cdot}A^{\tau(i)-\tau(j)}S_{\tau(j)-1}H_{\rho(j),\cdot}^\top + \delta_{\tau(i),\tau(j)}R_{\rho(i),\rho(j)},$$

for $\tau(i) \geq \tau(j)$, where $\tau()$ and $\rho()$ are the functions for which $Z_t^i = Y_{t+\tau(i)}^{\rho(i)}$, $H_{i,\cdot}$ indicates the ith row of H, $S_i = \sum_{k=1}^i A^{k-1}Q(A^{k-1})^\top$, and $R_{i,j}$ indicates the (i,j)th element of R. Otherwise, $\Phi_{ji} = \Phi_{ij}$.

Given this state mapping, it remains to be shown that there are values of the PLG parameters for which this state mapping is maintained by the update equations; this is done in the online appendix (Rudary & Singh 2008).

3 Consistent Estimation of PLG Parameters

Parameters of the LDS are often estimated using Expectation Maximization (EM) (Ghahramani & Hinton 1996)[6], an algorithm that aims to maximize the likelihood of the data by iteratively improving parameter estimates. EM suffers from problems with local optima, in part because there are many symmetries in the LDS parameter space.

To estimate the parameters of the PLG, we have modified the Consistent Estimation algorithm (CE) described by Rudary et al. (2005; 2006) to apply to the vector-valued PLG. An estimator is consistent if it converges in probability to the true value, where convergence in probability is defined as follows:

Definition 2. *The sequence $\widehat{x}_1, \widehat{x}_2, \ldots$ converges to x in probability if $\lim_{n\to\infty} \Pr(|\widehat{x}_n - x| > \delta) = 0$ for all positive δ. We write this as "$\widehat{x}_n \xrightarrow{p} x$ as $n \to \infty$."*

CE's estimates converge in probability to the true values of the parameters as the size of the training data set increases, avoiding problems with local optima.

3.1 The CE Algorithm

The CE algorithm is based in large part on the direct connection of the PLG parameters with statistical patterns in the data. For instance, G is the linear trend in the mean of the observations. This is in contrast to the LDS, in which the linear trend in the mean of the observations depends on both H and A. CE is noniterative and consists of simple operations like linear regression and taking sample means and covariances.

We assume that we have a data set consisting of K trajectories, each of which is a sequence of N observations. Each trajectory is started from the same (unknown) initial state μ_0, Σ_0.[7] The tth observation from the kth trajectory is denoted y_t^k; similarly, z_t^k is the observed value of Z_t from the kth trajectory (though this notation is the same as that used to indicate a specific element of y_t or z_t, the superscript k will always indicate the entire vector, taken from the kth trajectory). In addition, $\widehat{G}$ refers to an estimate of G (similarly $\widehat{C}_\eta$, $\widehat{\eta}_{t+1}$, etc.). The algorithm as given assumes that the dimension n is known and Z_t is prepopulated—i.e. that we know which observational element corresponds to each element of Z_t—and that all elements of Y_{t+1} are elements of Z_t, so that J is entirely predetermined.

We will break the presentation of this algorithm into three parts: estimating the linear trends ($G, \Gamma_1, \ldots, \Gamma_{\tau_{\max}}$), estimating the initial state (μ_0, Σ_0), and estimating the noise parameters (C_η, Σ_η).

Linear Trend Both G and $\Gamma_1, \ldots, \Gamma_{\tau_{\max}}$ represent linear trends in the data. G is the linear trend of Z_t in the absence of actions, and $\Gamma_1, \ldots, \Gamma_{\tau_{\max}}$ model the linear effects of the actions. CE estimates both together using a linear regression. However, because several rows of $G, \Gamma_1, \ldots, \Gamma_{\tau_{\max}}$ are determined by the makeup of Z_t, some care is required in setting up the regression.

Some rows of G copy an observation element from $i+1$ steps in the future onto the same element i steps in the future. For instance, if $Z_t^1 = Y_{t+1}^1$ and $Z_t^2 = Y_{t+2}^1$, the first row of G simply copies the second element of Z_t into the first. We refer to the unknown rows (i.e. those *not* predetermined in this way) as G^{unk}. In addition, the ith row of Γ_j (for each j) is fixed to be $\mathbf{0}$ when Z_t^i is not the first appearance of an observational element. Note that the *same rows* are predetermined in G and $\Gamma_1, \ldots, \Gamma_{\tau_{\max}}$. So by analogy, we refer to the rows that are *not* predetermined in $\Gamma_1, \ldots, \Gamma_{\tau_{\max}}$ as $\Gamma_1^{\mathrm{unk}}, \ldots, \Gamma_{\tau_{\max}}^{\mathrm{unk}}$.[8] Likewise, let $z_t^{\mathrm{unk},k}$ be the sub-vector of z_t^k that corresponds to these rows.

Let the notation $\overline{y}_t$ denote the average of y_t^k over all the trajectories (similarly $\overline{z}_t$, $\overline{\eta}_{t+1}$, etc.). From the PLG's core dynamics (2), it follows that $z_{t+1}^{\mathrm{unk},k} = G^{\mathrm{unk}}z_t^k + \sum_{i=1}^{\tau_{\max}} \Gamma_i^{\mathrm{unk}}u_{t+i}^k + \eta_{t+1}^k$ for all t and k. This equality holds when averaging across all trajectories in the data set:

$$\overline{z}_{t+1}^{\mathrm{unk}} = G^{\mathrm{unk}}\overline{z}_t + \sum_{i=1}^{\tau_{\max}} \Gamma_i^{\mathrm{unk}}\overline{u}_{t+i} + \overline{\eta}_{t+1}$$

[6] Though see Section 4 for a discussion of an alternative approach, 4SID.

[7] Note that in the LDS case, this is equivalent to requiring that $X_1 \sim \mathcal{N}(x_1^-, P_1^-)$ for some x_1^- and P_1^-, *not* that $X_1 = x_1$ for some x_1; this is just putting a prior on the initial observations.

[8] However, some of the rows of $\Gamma_1^{\mathrm{unk}}, \ldots, \Gamma_{\tau_{\max}}^{\mathrm{unk}}$ are also fixed at $\mathbf{0}$. This is to preserve causality, as discussed in Section 2—actions should not affect observations that will occur before the action is taken. This will be taken into account in the regression.

for all t. As K grows large, the Weak Law of Large Numbers applies, and it can be shown that $\overline{\eta}_{t+1} \xrightarrow{p} \mathbf{0}$ as $K \to \infty$, where $\xrightarrow{p}$ denotes convergence in probability (defined below). This then leaves the equation

$$\overline{z}^{\text{unk}}_{t+1} \xrightarrow{p} \gamma \xi_t \tag{8}$$

for all t, where $\gamma \triangleq \begin{pmatrix} G^{\text{unk}} & \Gamma^{\text{unk}}_1 & \cdots & \Gamma^{\text{unk}}_{\tau_{\max}} \end{pmatrix}$ and

$$\xi_t^\top \triangleq \begin{pmatrix} \overline{z}_t^\top & \overline{u}_{t+1}^\top & \overline{u}_{t+2}^\top & \cdots & \overline{u}_{t+\tau_{\max}}^\top \end{pmatrix}.$$

(8) can be rewritten in matrix form as $Z^{\text{unk}} \xrightarrow{p} \gamma \Xi$, where the tth column of Ξ is ξ_{t-1} and the tth column of Z^{unk} is $\overline{z}^{\text{unk}}_t$; both of these matrices have $N - \tau_{\max}$ columns. This is a rather straightforward linear regression problem. If it weren't for some of the elements of γ being fixed at 0, the solution would be

$$\widehat{\gamma} = Z^{\text{unk}} \Xi^\top (\Xi \Xi^\top)^{-1} \xrightarrow{p} \gamma.$$

As it is, each row of $\widehat{\gamma}$ can be computed using a separate regression, where the columns of Ξ corresponding to the zero-fixed elements are removed. Having computed an estimate of G^{unk} and $\Gamma^{\text{unk}}_1, \ldots, \Gamma^{\text{unk}}_{\tau_{\max}}$, we can combine that with the rows predetermined by the structure of the model to obtain $\widehat{G}$ and $\widehat{\Gamma}_1, \ldots, \widehat{\Gamma}_{\tau_{\max}}$, estimates of G and $\Gamma_1, \ldots, \Gamma_{\tau_{\max}}$, respectively. These are, in general, biased estimates of G and $\Gamma_1, \ldots, \Gamma_{\tau_{\max}}$, as the noise in the rows of Z^{unk} are not independent of one another. However, we have shown that this regression produces a consistent estimate—that is, one that converges in probability on the correct answer.

Initial State At first blush, it may seem that the initial state μ_0 and Σ_0 could be estimated through a sample mean and sample covariance of z^k_0, respectively. However, this fails to account for the actions; the initial state is the distribution of Z_0 given that $u_1 = \mathbf{0}, u_2 = \mathbf{0}$, etc. This can be remedied by noting the effect of each action $u_1, u_2, \ldots, u_{\tau_{\max}}$ on Z_0 is additive and a function of only G, $\Gamma_1, \ldots, \Gamma_{\tau_{\max}}$, and the action itself. By using the *estimates* $\widehat{G}$, $\widehat{\Gamma}_1, \ldots, \widehat{\Gamma}_{\tau_{\max}}$ to compute this effect, an estimate of $Z_0 | u_1 = \mathbf{0}, \cdots$ can be obtained for each trajectory, and the sample mean and covariance of these estimates of Z_0 can be taken to estimate μ_0 and Σ_0. Algebraic manipulation produces the result that these estimates can be obtained by $\widehat{z}^k_{0|0} = z^k_0 - \sum_{i=0}^{\tau_{\max}-1} \sum_{j=i}^{\tau_{\max}-1} (\widehat{G}^{j-i} \widehat{\Gamma}_j) u^k_i$, where it can be assumed that $u^k_0 \triangleq \mathbf{0}$.

Given the estimates $\widehat{z}^k_{0|0}$, the initial state can be estimated through a sample mean and variance:

$$\widehat{\mu}_0 = \frac{1}{K} \sum_{k=1}^{K} \widehat{z}^k_{0|0},$$

$$\widehat{\Sigma}_0 = \frac{1}{K} \sum_{k=1}^{K} (\widehat{z}^k_{0|0} - \widehat{\mu}_0)(\widehat{z}^k_{0|0} - \widehat{\mu}_0)^\top.$$

These are consistent estimates of the initial state.

Noise Parameters The variance Σ_η of the noise and covariance C_η of the noise with Z_t can be estimated similarly. First, we estimate the noise terms by rewriting (2):

$$\widehat{\eta}^k_{t+1} = z^k_{t+1} - \widehat{G} z^k_t - \sum_{i=1}^{\tau_{\max}} \Gamma_i u^k_{t+i} \qquad \forall k, t.$$

Knowing that $\mathrm{E}[\eta_{t+1}] = 0$, the sample covariance calculations yield

$$\widehat{\Sigma}_\eta = \frac{1}{K(N - \tau_{\max})} \sum_{t=1}^{N-\tau_{\max}} \sum_{k=1}^{K} \widehat{\eta}^k_t \widehat{\eta}^{k\top}_t \quad \text{and}$$

$$\widehat{C}_\eta = \frac{1}{K(N - \tau_{\max})} \sum_{t=0}^{N-\tau_{\max}-1} \sum_{k=1}^{K} z^k_t \widehat{\eta}^{k\top}_{t+1}.$$

Consistency The estimates $\widehat{G}, \widehat{\Gamma}_1, \ldots, \widehat{\Gamma}_{\tau_{\max}}, \widehat{\mu}_0, \widehat{\Sigma}_0, \widehat{\Sigma}_\eta$, and $\widehat{C}_\eta$ are consistent; that is, as the number of trajectories in the data set increases, the estimates converge in probability to the true values.

There are some technical requirements on the system and the policy used to generate the data set that must be satisfied in order for the consistency result to hold. The major requirement is that the matrix $\Xi \Xi^\top$ be invertible in the limit as $K \to \infty$. This is both a theoretical and a *practical* requirement. It is a theoretical requirement because consistency is a property of the behavior of estimators in the limit. It is practical because, when this condition does *not* hold, conditioning problems will lead to poor estimates of G and $\Gamma_1, \ldots, \Gamma_{\tau_{\max}}$ on large data sets. Since most of the other estimates *depend* on the estimates for these parameters, this well lead to a poor model overall.

An equivalent way of stating this requirement is that a) $\mathrm{E}[\Xi]$ have full rank and b) the trajectories have sufficient length to make Ξ at least as wide as it is tall. The latter requirement means that $N \geq n + (l+1)\tau_{\max}$. The former is a joint constraint on the *system* and the *policy* used to generate the data set. In an uncontrolled system, this condition will not be satisfied if, for example, $\mu_0 = \mathbf{0}$ or μ_0 is an eigenvector of G. In a *controlled* system, the actions affect $\overline{z}_t$, so a well-chosen policy can overcome difficulties such as $\mu_0 = \mathbf{0}$. However, a new issue is raised: The actions form part of Ξ, so a random policy where $\mathrm{E}[u_1] = \mathrm{E}[u_2] = \cdots$ will lead to an under-ranked matrix. An exploration policy that satisfies the requirement that $\mathrm{E}[\Xi]$ have full rank for a given system is called a *CE-learnable* policy for that system.

Theorem 2. *If a dynamical system can be modeled by an n-dimensional PLG, is CE-learnable, and generates a training set whose trajectories are at least $n + (l+1)\tau_{\max}$ time steps long, then, as the number of trajectories K grows, the parameter estimates computed by the CE algorithm will converge in probability to the true parameters.*

This is proven in an online appendix (Rudary & Singh 2008).

4 Related Work

There are two areas of related research that should be compared to this work. Here we will briefly contrast our work to

that of Jaeger (2001), and then compare our CE estimation algorithm to subspace identification methods in LDSs.

Observable Operator Models (OOMs) are a general class of models in which observations are modeled as operators on a state vector (Jaeger 1997). Some OOMs are predictive state models in that the elements of the state vector correspond to predictions about future observations. The OOM discussed by Jaeger (2001) is related to a hidden Markov model that emits observations from a continuous distribution conditioned on the value of a discrete latent process. The structure of the model is quite different from that of the PLG. The continuous OOM is a "blending" of discrete OOMs that uses the unobserved outputs of a discrete OOM to generate a probability distribution over observable values; it is a generalization of the *semi-continuous HMM* (Bengio 1999). Thus there is a latent component to this continuous OOM that is not present in the PLG. As a result of these differences, the systems that can be modeled by this OOM are different than those that can be modeled by the PLG.

Subspace-based state-space system identification (4SID) methods are parameter estimation algorithms for LDSs (Viberg 1995). Like CE, 4SID methods are noniterative—in particular, they do not perform local searches in the parameter space, as EM methods do. These methods can be complicated, here I present a basic intuition. The core dynamics of the LDS can be written as

$$\begin{pmatrix} X_{t+1} \\ Y_t \end{pmatrix} = \begin{pmatrix} A \\ H \end{pmatrix} X_t + \begin{pmatrix} \omega_{t+1} \\ \nu_t \end{pmatrix}.$$

This is quite similar to (2); if an estimate for X_t were obtained, A and H could be estimated through least-squares linear regression, much as G is in the CE algorithm. The main question answered by subspace identification is how to estimate X_t. Define the output block Hankel matrices

$$Y_p = \begin{pmatrix} y_1 & \cdots & y_j \\ \vdots & \ddots & \vdots \\ y_i & \cdots & y_{i+j-1} \end{pmatrix}, \; Y_f = \begin{pmatrix} y_{i+1} & \cdots & y_{i+j} \\ \vdots & \ddots & \vdots \\ y_{2i} & \cdots & y_{2i+j-1} \end{pmatrix},$$

where f stands for future and p for past, i is at least n, and j is chosen so that the entire data set is included in Y_p and Y_f (4SID methods operate on a single trajectory). Define $\widehat{\mathbf{X}}_{i+1} = [\widehat{x}_{i+1} \; \ldots \; \widehat{x}_{i+j}]$ as the estimates of the latent variables $\mathbf{X}_{i+1} = [X_{i+1} \; \cdots \; X_{i+j}]$. Finally, let Γ_i be the extended observability matrix

$$\Gamma_i = \begin{pmatrix} H^\top \\ (HA)^\top \\ \cdots \\ (HA^{i-1})^\top \end{pmatrix}.$$

By definition, $\mathrm{E}[Y_f | \mathbf{X}_{i+1} = \widehat{\mathbf{X}}_{i+1}] = \Gamma_i \widehat{\mathbf{X}}_{i+1}$. But it can also be shown that $Y_f/Y_p \xrightarrow{p} \Gamma_i \widehat{\mathbf{X}}_{i+1}$ under certain conditions as $j \to \infty$, where $Y_f/Y_p = Y_f Y_p^\top (Y_p Y_p^\top)^\dagger Y_p$ is the projection of the row space of Y_f onto the row space of Y_p and $A^\dagger$ is the Moore-Penrose pseudo-inverse of the matrix A. Γ_i can be recovered through a singular value decomposition of Y_f/Y_p, and so $\widehat{\mathbf{X}}_{i+1}$ can be computed by $\widehat{\mathbf{X}}_{i+1} = \Gamma_i^\dagger (Y_f/Y_p)$.

4SID methods vary in their details—for instance, $\widehat{\mathbf{X}}_{i+1}$ may not be computed explicitly, and other changes may be made for the sake of efficiency or statistical guarantees. Some variants of 4SID are consistent under quite general conditions.

One difference between CE and 4SID methods is that 4SID methods are consistent as the length of a single trajectory grows, while CE is consistent as the number of trajectories grows. Some problems are naturally episodic, and so it is easier to produce a data set with a large number of trajectories instead of a single trajectory. For example, when trying to model the behavior of other drivers on the road, it makes more sense to observe many vehicles' behavior and treat each vehicle's trajectory separately. Of course, the opposite is also true: In some problems, it is easier to obtain long trajectories than many trajectories. This difference make it quite difficult to compare the two algorithms with any precision because they work on different kinds of data sets.

5 Conclusion

We have introduced an improved Predictive Linear Gaussian model, a model for stochastic dynamical systems with vector-valued observations. We have also introduced a variance-adjusted version of the model. We have shown an equivalence between n-dimensional variance-adjusted PLGs and n-dimensional Linear Dynamical Systems. Finally, we have introduced a consistent parameter estimation algorithm for PLGs.

This work significantly improves the state of the art in predictive state representations, for the first time allowing them to handle real-vector-valued observations, a feature of many realistic problems. It also adds to the field of linear dynamical systems by introducing another consistent estimation algorithm for a model that subsumes them.

However, there is much room for future work. One future direction is to develop a maximum likelihood estimation algorithm for PLGs. We are also working on a single-trajectory CE algorithm; besides allowing parameter estimation on data sets containing a single long trajectory, it will also allow us to compare CE to subspace identification methods experimentally.

References

Bengio, Y. 1999. Markovian models for sequential data. *Neural Computing Surveys* 2:129–162.

Catlin, D. E. 1989. *Estimation, control, and the discrete Kalman filter*. Springer-Verlag New York.

Ghahramani, Z., and Hinton, G. E. 1996. Parameter estimation for linear dynamical systems. Technical report, Dept. of Computer Science, U. of Toronto.

Jaeger, H. 1997. Observable operator models II: Interpretable models and model inductions. Technical Report 1083, GMD, St. Augustine.

Jaeger, H. 2001. Modeling and learning continuous-valued stochastic processes with OOMs. Technical Report 102, GMD, St. Augustine.

Kalman, R. E. 1960. A new approach to linear filtering and prediction problem. *Transactions of the ASME—Journal of Basic Engineering* 82(Series D):35–45.

Littman, M. L.; Sutton, R. S.; and Singh, S. 2001. Predictive representations of state. In *NIPS 14*.

Pandit, S. M., and Wu, S.-M. 1983. *Time Series and System Analysis With Applications*. Krieger Publishing Company.

Rudary, M., and Singh, S. 2006. Predictive linear-Gaussian models of controlled stochastic dynamical systems. In *ICML 2006*.

Rudary, M., and Singh, S. 2008. Proofs for PLG models of vector-valued dynamical systems. In *Online appendix*. http://www.eecs.umich.edu/˜mrudary/isaim08-supplement.pdf.

Rudary, M.; Singh, S.; and Wingate, D. 2005. Predictive linear-Gaussian models of stochastic dynamical systems. In *UAI 21*.

Singh, S.; James, M. R.; and Rudary, M. 2004. Predictive state representations: A new theory for modeling dynamical systems. In *UAI 20*.

Viberg, M. 1995. Subspace-based methods for the identification of linear time-invariant algorithms. *Automatica* 31(12):1835–1851.

Welch, G., and Bishop, G. 2004. An introduction to the Kalman filter. Technical Report 95-041, U. of N. Carolina at Chapel Hill, Dept. of Computer Science.

Wolfe, B.; James, M. R.; and Singh, S. 2005. Learning predictive state representations in dynamical systems without reset. In *ICML 22*.