Experimental design for Partially Observed Markov Decision Processes

Leifur Thorbergsson, Giles Hooker *

Department of Statistics, Cornell University, Ithaca, NY 14850

May 5, 2014

Abstract

This paper deals with the question of how to most effectively conduct experiments in Partially Observed Markov Decision Processes so as to provide data that is most informative about a parameter of interest. Methods from Markov decision processes, especially dynamic programming, are introduced and then used in an algorithm to maximize a relevant Fisher Information. The algorithm is then applied to two POMDP examples. The methods developed can also be applied to stochastic dynamical systems, by suitable discretization, and we consequently show what control policies look like in the Morris-Lecar Neuron model, and simulation results are presented. We discuss how parameter dependence within these methods can be dealt with by the use of priors, and develop tools to update control policies online. This is demonstrated in another stochastic dynamical system describing growth dynamics of DNA template in a PCR model.

1 Introduction

Hidden Markov Models have proven their usefulness across a wide variety of applications. In many of these applications, the user or the experimenter

*Leifur Thorbergsson is a graduate student at the Department of Statistical Science, Cornell University, Ithaca, NY 14850 (email: lt274@cornell.edu); and Giles Hooker is an Associate Professor at the same department (email: giles.hooker@cornell.edu).
will have some way of influencing the transitions of the underlying Markov Chain, as in Markov Decision Processes, and such a process is called a Partially Observed Markov Decision Process (POMDP), see Monahan [5]. If we assume that the transition probability matrix is governed by some unknown parameters, an important problem is to understand how the process can be influenced to get data that is most informative about the parameters. We can think of this as experimental design for Partially Observed Markov Decision Processes.

We consider a POMDP \((x_t, y_t, u_t)_{t=0,...,T}\). In this setting \(x_t\) is an unobserved Markov Chain, where the transition probabilities depend in a parametric way on what control \(u_t\) is chosen at time \(t\) and an unknown parameter \(\theta\). The process \(y_t\) is observed and depends on which state \(x_t\) is in.

Our goal is to find ways to use the controls \(u_t\) to improve parameter estimates of \(\theta\). Since the maximum likelihood estimates for \(\theta\) will be asymptotically efficient, our general strategy will be to use the controls to try to minimize the sample variance of the maximum likelihood estimates of \(\theta\). This will be achieved by maximizing a Fisher Information for \(\theta\). The controls are calculated using dynamic programming, a popular maximization algorithm from Markov Decision Processes which outputs an adaptive control policy, i.e. the control chosen at time \(t\) is based on observations up to time \(t\).

The first attempt at using dynamic controls to maximize a Fisher Information was by Lin et al. [4] who proposed maximizing the Fisher Information that corresponds to direct observations of the underlying process \(x_t\), labeled the Full Information Fisher Information (FOFI), and using a filter to compute \(x_t\) if it is not observed directly. This paper extends their work by making use of the POMDP structure and proposes maximizing a Fisher Information that is based on the observations \(y_t\), labeled the Partial Observation Fisher Information (POFI). The control policies based on these two Fisher Informations, FOFI and POFI, are compared, and we illustrate a setting in which the two control policies are quite different.

The methods we use to calculate controls for maximizing Fisher Information will depend on the unknown parameter \(\theta\). We illustrate how this problem can partially be overcome by assuming a prior for \(\theta\) to calculate a control policy before running the experiment. Additionally we describe how, using data acquired as the experiment progresses, a posterior for \(\theta\) can be used to calculate a more precise control policy. That is, parameter information from observations acquired at a time \(t\) can be used to improve the policy used in what is left of the experiment. These methods will be based
on the Value Iteration Algorithm (VIA), which is closely related to dynamic programming.

The methods developed have application value beyond Partially Observed Markov Decision Processes. Lin et al. [4] considered stochastic systems of the form

\[ dx = f(x, \theta, u(t))dt + \Sigma^{1/2}dW \]

where \( \theta \) is the parameter of interest, to be estimated, \( u(t) \) is a control that can be chosen by the user, \( x \) is the vector of state variables, \( f \) is a vector valued function, \( W \) a Wiener process, and additionally \( x(t) \) is only observed partially or noisly. By discretizing time, state and observation spaces the process can be approximated by a POMDP, allowing us to use the methods developed in this paper to devise a control policy that maximizes information about the parameter \( \theta \).

In order to illustrate our methods we present four examples, ordered by level of complexity, with the first two being POMDP’s and the latter two continuous stochastic systems. We use the unknown \( \theta \) to calculate controls in the first three examples to highlight the differences between POFI and FOFI, but in the last example we examine means to deal with the dependence of the POFI and FOFI policies on the parameter of interest.

In the first example we hypothesize about the kind of systems in which we will observe large improvement in parameter estimation by using the POFI control policy over the FOFI policy. Following a discussion we construct a mock Partially Observed Markov Decision Process, in which this improvement is shown using a simulation study.

To illustrate the real-world applicability of design in discrete POMDP’s we consider a realistic POMDP from experimental economics. The model will consist of a simple adversarial game similar to the “rock - paper - scissor” game where one player tries to play in such a way that maximizes information about the other players’ strategy.

The third example is a stochastic version of the Morris-Lecar Neuron model, a dynamical system which models voltage in a single neural cell. This model is two dimensional, but only one dimension is observed. The model has multiple parameters and we investigate how the POFI and FOFI control policies perform in estimating them.

Fourthly we consider an example from biology, a Polymerase chain reaction (PCR) experiment where DNA template is grown in liquid substrate. The population dynamics are modeled in a dynamical system with stochastic
errors, and the aim is to estimate the half-saturation constant, a parameter which controls the saturation of the template. Here we compare using a prior for $\theta$ and using VIA to calculate a control policy.

In Section 2 we describe the POMDP setup, and then introduce dynamic programming, an algorithm from Markov Decision Processes. This algorithm is then applied to evaluate control policies that maximize both the Partial Observation Fisher Information and the Full Observation Fisher Information.

In Section 3 we show how the control policies can be applied to POMDP examples and then, in Section 4, describe how stochastic dynamical systems can be transformed to POMDPs by using discretization methods, and give an example of such a system. In Section 5 we discuss the problem of parameter dependence when running the dynamic programs, and how these issues can be dealt with. This is illustrated in the last example, and a discussion follows.

2 Framework and Assumptions

We consider a Markov decision process $(X_t, u_t)_{t=0,\ldots,T}$. In this setting $X_t$ is a Markov chain, but the transition probabilities at time $t$ depend on a control $u_t$ chosen at that time. We assume a finite state space $\mathcal{X}$ for the state process $X_t$ and that the controls available belong to some finite set $\mathcal{U}$. We let $K$ denote the size of $\mathcal{X}$ and $l$ the size of $\mathcal{U}$. The transition probabilities are assumed to be parametric and we write $p(x_{t+1}|x_t, u_t, \theta)$ short for $p(x_{t+1} = x^i|x_t = x^j, u_t = u^r, \theta)$ where $x^i, x^j \in \mathcal{X}$ and $u^r \in \mathcal{U}$.

In addition to this we assume that the process $X_t$ is latent and we only observe the related observations $Y_t \in \mathcal{Y}$ whose relation to the $X_t$ can also depend on $\theta$. We write $p(y_t|x_t, \theta)$ short for $p(y_t = y^i|x_t = x^j, \theta)$, where $x^j \in \mathcal{X}$ and $y^i \in \mathcal{Y}$, and let $L$ denote the size of $\mathcal{Y}$. This makes the system a Partially Observed Markov Decision Process. It has a finite horizon $T$ in which we observe $y_0 \ldots y_T$. We will use the short hand notation $y_{m:t}$ to denote $y_m, \ldots, y_t$, i.e. the observations between time $m$ and $t$, and analogous notation for $u_t$ and $x_t$.

The objective is to use the controls to maximize the information we get about the parameter $\theta$ through the observed process $y_{0:T}$. The parameter estimation is done using maximum likelihood and it is therefore natural to try to maximize the Fisher Information of our observed process which we can
express as

\[ FI(\theta) = E \left[ \sum_{t=0}^{T-1} \left( \frac{\partial}{\partial \theta} \log p(y_{t+1}|y_{0:t}, u_{0:t}, x_0, \theta) \right)^2 \right] \]

which we label as the Partial Observation Fisher Information (POFI), see Appendix A.2 for details on its construction. When we consider continuous time dynamical systems the observation spaces will be continuous, but we will use this discretized Fisher Information as an approximation to the actual Fisher Information of the observations.

2.1 A Dynamic Program

In order to maximize the Fisher Information we employ the techniques of stochastic dynamic programming. Assuming a reward function \( C_t(x_t, u_t) \), our objective is to maximize the total expected reward \( E[\sum_{t=0}^{T} C_t(X_t, u_t)] \) by use of the controls. The essence of dynamic programming is that by starting at time \( T - 1 \) and working backwards, we can compute an optimal policy that maps a state \( x_t \) to a control \( u_t \) that accounts for the choices of \( u_t \) that we will make in the future.

In a generic dynamic program we set \( V_T = 0 \) and and then going backwards from \( t = T - 1, \ldots, 0 \) solve

\[ V_t(x_t) = \max_{u_t} \{ E_{x_{t+1}}[C_t(x_t, u_t) + V_{t+1}(x_{t+1})] \} \]

where \( V_t \) is called the value function, and we get the associated control

\[ u_t^*(x_t) = \arg\max_{u_t} \{ E_{x_{t+1}}[C_t(x_t, u_t) + V_{t+1}(x_{t+1})] \} \]

for every state \( x_t \). This will give us a policy of what control to use at a certain state \( x_t \) at a certain time \( t \). The use of these controls will maximize the total expected reward \( E[\sum_t C_t(X_t, u_t)] \). We refer to Puterman [7] for a detailed description of dynamic programming.

With this in mind, we set \( C_t(y_{0:t}, u_{0:t}, \theta) = \left( \frac{\partial}{\partial \theta} \log p(y_{t+1}|y_{0:t}, u_{0:t}, x_0, \theta) \right)^2 \) and we try to maximize the Fisher Information \( FI(\theta) = E[\sum_t C_t(y_{0:t}, u_{0:t}, \theta)] \). Note that in this instance the reward function depends on the entire history of observations and controls up to time \( t \).

5
The Value function in the corresponding dynamic program is

\[ FI_t(y_{0:t}, u_{0:t-1}, \theta) = \max_{u_t} \left\{ E_{y_{t+1}} [C_t(y_{0:t}, u_{0:t}, \theta) + FI_{t+1}(y_{0:t+1}, u_{0:t}, \theta)] \right\} \]

and we denote it the Fisher Information to Go.

A problem here is that just in the first step of the dynamic program \( t = T - 1 \) we would have to calculate the Fisher Information to Go for \( L_{T-1}^T \) many combinations of \( y_{0:t} \) and \( u_{0:t-1} \). This is formidable for even modest dimensions. We therefore approximate the process by conditioning only on the last \( m + 1 \) observations in the Fisher Information:

\[ FI_{approx} = E \sum_{t=0}^{T-1} \left( \frac{\partial}{\partial \theta} \log p(y_{t+1} | y_{t-m:t}, u_{t-m:t}, \nu_{t-m}, \theta) \right)^2 \]

where \( \nu_{t-m} \) is some prior that we assume for \( x_{t-m} \), although we generally suppress it in notation since we assume it is fixed. If \( t - m < 0 \) we set \( t - m : t \) to mean \( 0 : t \) to ease notation.

The reward becomes \( C(y_{t-m:t}, u_{t-m:t}, \theta) = \left( \frac{\partial}{\partial \theta} \log p(y_{t+1} | y_{t-m:t}, u_{t-m:t}, \theta) \right)^2 \)

and \( FI_{t:m}(y_{t-m:t}, u_{t-m:t-1}, \theta) = \max_{u_t} \left\{ E_{y_{t+1}} [C + FI_{t+1,m}(y_{t-m:t}, u_{t-m:t}, \theta)] \right\} \)

the Fisher Information To Go. The pseudocode for the corresponding dynamic program is given in Appendix A.1.1. For this approximate dynamic program to be sensible we want the approximated Fisher Information to approach the true Fisher Information as \( m \) increases. This holds given that the POMDP process satisfies certain technical mixing conditions, which we have stated in detail in Assumption 1 in Appendix A.3.

**Theorem 1.** Assume the conditions in Assumption 1 hold. Then, for \( m < T \) and any control policy, we have

\[ |FI - FI_{approx}| \leq c_1 (T - 1 - m) \rho^{m/2} \]

where the constant \( c_1 \) and \( \rho \) do not depend on \( m \) or \( T \), see Appendix A.4.

The proof requires extensions of work in Cappe et al. [1] and is given in Appendix A.4 with further technical results reserved to Appendix 2. Theorem 1 states that \( FI_{approx} \) approaches the true Fisher Information exponentially as \( m \) increases, and is thus a viable approximation for the Fisher Information in the dynamic program.

The runtime of the dynamic program however also grows exponentially in \( m \) and we found that while setting \( m = 0 \), i.e. conditioning on one
observation, gave poor results in some of our simulations, conditioning on
two observations, i.e. $m = 1$, generally gave good results when compared to
other control policies. Setting $m = 2$ increased runtime greatly and was in
most of our applications infeasible without making more approximations to
how the dynamic program is run. The exact effect of increasing $m$ is problem
specific, and a thorough analysis of what $m$ to choose is beyond the scope of
this paper.

2.2 FOFI Dynamic Program

An alternative method to choose controls was proposed by Lin et al. [4]. They
considered constructing an optimal control policy for the Fisher Information
that would apply if $(X_t)$ were observed directly;

$$FI = E \sum_{t=0}^{T-1} \left( \frac{\partial}{\partial \theta} \log p(x_{t+1}|x_t, u_t, \theta) \right)^2$$

This is labeled the Full Observation Fisher Information (FOFI). As noted
before, when considering continuous time stochastic systems, the state space
is continuous, but we use this Fisher Information as an approximation to
the continuous state Fisher Information. An advantage of using FOFI over
POFI is that when running the dynamic program the Markov property of the
Markov Decision Process $(X_t, u_t)$ allows us to only consider a maximization
over the state space $x_t \in \mathcal{X}$ but not past values $x_{0:t-1}$. The dynamic program
for FOFI is given in Appendix A.1.2.

However, maximizing FOFI can lead to suboptimal controls since it is
not the correct Fisher Information for the data. Additionally, when the
actual experiment is run we do not observe $X_t$. Instead we have to use the
observed values to get a probability distribution (a filter) on the state $x_t,
p(x_t|y_{0:t}, u_{0:t-1}, x_0, \theta)$ and use the control associated with the state that has
the highest probability.

The computation cost of running a dynamic program with FOFI is $O(TK^2l)$,
generally lower than that of POFI; $O(TL^{m+1}l^{m+1})$. The cost of estimating
the state $X_t$ at runtime is $O(K^2)$ at each time point $t$. For details see B.1 in
Appendix 2.
2.3 Parameter Estimation

After running an experiment, using one of the control policies, the parameter $\theta$ is estimated either via an EM algorithm or by directly maximizing the loglikelihood. These two estimation methods had similar accuracy in our simulations, although the EM algorithm was generally slower. Convergence of the EM algorithm is discussed in Cappe et al. [1] for Hidden Markov Models and extends naturally to POMDP’s.

For the asymptotic properties of the MLE we refer to Cappe as well [1], where conditions for consistency and asymptotic normality in Hidden Markov Models are given. The central elements of their proof are the stationarity of the process $(X_t, Y_t)$ along with certain forgetting properties of the filter, meaning that ignoring all but the past $m$ observations (as we do) yields an error in the filter distribution that decreases exponentially in $m$. We note that if we employ a time-independent control policy (as we do in Section 5), we obtain to a Hidden Markov Model and can rely on [1] if we assume stationarity. In Appendix 2 we establish extensions of forgetting properties for POMDP models more generally, which points to a more general asymptotic theory for the MLE in this case, but do not pursue this here.

Theorem 1 shows that using $FI_{\text{approx}}$ is a good approximation to the Partial Observation Fisher Information for running a dynamic program. This provides a control policy that is an approximation to the optimal control policy. Now consider using this approximate policy to run an experiment and then estimating $\theta$ by evaluating the MLE. The asymptotic variance of this MLE will be the the inverse of the Partial Observation Fisher Information, with controls from the approximate policy. It is therefore of interest to compare POFI with an optimal policy and POFI with the approximate policy. In Appendix A.5 we show

**Theorem 2.** Given that the mixing conditions in Assumption 1 (A.3) hold we have

$$0 \leq FI(u_0^*, \ldots, u_{T-1}^*) - FI(u_{0,m}^*, \ldots, u_{T-1,m}^*) \leq c_2T(T+1)p^{m/2}$$

where $u_0^*, \ldots, u_{T-1}^*$ are the optimal controls, $u_{0,m}^*, \ldots, u_{T-1,m}^*$ the approximate optimal controls and $FI$ is the Partial Observation Fisher Information. The constant $c_2$ and $p$ do not depend on $m$ or $T$, see Appendix A.5 for details.
That the asymptotic variance of the MLE converges to the best possible asymptotic variance exponentially quickly in \( m \) further supports our approximations.

3 Discrete Examples

3.1 6 state example

While the FOFI strategy has been shown to be effective in Lin et al. [4] it is possible to define systems in which the strategy is not optimal and may in fact be worse than just using fixed or random controls. Usually certain parts of state space will give more information about a parameter than others, given that the state space is perfectly observed. In these cases optimal controls would try to move the process to these states. However, if the state space is only partially observed, most information might be obtained in different parts of state space and the FOFI controls become suboptimal. In cases like this POFI often does better than FOFI. In this example, we demonstrate a system where FOFI and POFI choose very different controls, and using a simulation study, we show that the controls chosen by POFI produce less variable parameter estimates.

Consider a discrete time Markov chain \( x_t \) with state space \( S_x = \{1, 2, 3\} \) and a transition probability matrix

\[
P = \begin{bmatrix}
\frac{1}{2} - \frac{p}{4} + \frac{u}{4} & \frac{1}{2} & .4 - \frac{u}{4} \\
\frac{1}{2} - \frac{p}{4} - \frac{u}{4} & \frac{1}{2} & .15 \\
\frac{1}{2} - \frac{p}{4} - \frac{u}{4} & \frac{1}{2} & .45 + \frac{u}{4}
\end{bmatrix}
\]

where the parameter of interest is \( p \in [0, .5] \) and the control is \( u \in \{-1, 1\} \). For \( x_t = 1 \) or \( x_t = 3 \), choosing the control \( u = 1 \) will increase the probability of the Markov chain staying in its current state while choosing \( u = -1 \) will increase the probability of it leaving its state.

Now assume this process isn’t observed directly but through a related process \( y_t \) with state space \( S_y = \{1, 2\} \) whose transition probabilities depend on which state \( x_t \) is in. We denote the transition probability matrices with \( (P_k)_{i,j} = p(y_{t+1} = j | y_t = i, x_t = k) \) given by

\[
P_1 = \begin{bmatrix}
.5 & .5 \\
.5 & .5
\end{bmatrix}, \quad P_2 = \begin{bmatrix}
.5 & .5 \\
.5 & .5
\end{bmatrix}, \quad P_3 = \begin{bmatrix}
1 - \frac{p}{2} & \frac{p}{2} \\
\frac{p}{2} & 1 - \frac{p}{2}
\end{bmatrix}
\]
Table 1: Long run control policy that results from using POFI in the 6 state example. The first column describes which control to use for a given history \((y_t, y_{t-1}, u_{t-1})\) of observations and control.

If \(x_t\) were observed we would get information about the parameter \(p\) when \(x_t = 3\) and from \(y_t\) when \(x_t = 3\). The idea here is that since the FOFI controls assume the whole state space is observed they might encourage \(x_t\) to be in state 1, while the POFI controls that take into account what is actually observed might choose the controls more intelligently. Indeed when calculating the controls according to FOFI the long run control is to “leave one’s state” if \(x_t = 3\) and “stay in one’s state” if \(x_t = 1\). It is harder to predict what kind of controls result from using POFI, but the control policy is given in Table 1.

To illustrate this difference, a simulation study was carried out to test which method performed better. The process \(x_t\) was run for 1000 steps with \(p = .37\), using controls chosen by POFI and again using those chosen by FOFI. Additionally we ran a simulation of the same length, but where the control was chosen randomly, with \(u = -1\) and \(u = 1\) having equal probability. Then the parameter \(p\) was estimated using an EM algorithm. This was done 500 times to get an empirical distribution for the estimates of \(p\). The results are given in Table 2. Estimates of \(p\) using a POFI policy had the lowest MSE and variability, but estimates based on the random policy had lower bias. Estimates using a FOFI policy were comparatively worse.

3.2 Adversarial game - A POMDP example

The following example describes an application of the above methods in the context of experimental economics. The problem is derived from Sachat et. al. [8], in which we wish to model how humans change their game-playing strategies over time.

We set up a game with two players: a Row player and a Column player.
Table 2: Simulation results for the 6 state example. We see that the controls chosen by POFI make for more accurate estimates of $p$. The FOFI policy does worse than a random policy.

|       | bias  | st. dev. | MSE  |
|-------|-------|----------|------|
| FOFI  | 0.0109| 0.1098   | 0.0121|
| POFI  | 0.0094| 0.0867   | 0.0076|
| Random| 0.0059| 0.0941   | 0.0089|

Table 3: Rewards in the Gamble Safe game. The first number is the reward for the Row player and the second number the reward for the Column player, given a certain outcome.

|       | Left | Right |
|-------|------|-------|
| Left  | 2.0  | 0.1   |
| Right | 1.2  | 1.1   |

They repeatedly play a game where both simultaneously choose either left or right, and they get rewards depending on the outcome, according to Table 3 the Row player would for example get 2 and the Column player 0 if both chose left. We follow [8] and assume that at any given play the Column player follows one of two strategies: the Nash-equilibrium strategy of choosing either left or right with 50% probability or the Gamble-safe strategy, where they only choose right. The player will pick either strategy based on a multinomial logistic model, where the probabilities depend on the last two plays of the Row player, and the last strategy chosen by the Column player. This results in a Partially Observed Markov Decision Process with the strategy employed being a hidden state giving rise to observed plays.

Let $S_t$ denote the strategy chosen by the Column player at time $t$, $U_t$ denote the action played by the Row player at time $t$. Let $S_t = -1$ if the Nash-equilibrium is chosen, $S_t = 1$ if the Gamble-safe strategy is chosen. Also let $U_t = 1$ if the Row player plays right, $U_t = -1$ if he plays left. Similarly $Y_t$ will denote the plays of the Column player. The strategy $S_{t+1}$ chosen at time $t + 1$ will then be chosen according to

$$P(S_{t+1} = -1) = \frac{e^x}{1 + e^x} \quad \text{and} \quad P(S_{t+1} = 1) = \frac{1}{1 + e^x}$$
where we let

\[ x = 1.2U_t + U_{t-1} + \theta S_t \]

The experiment is set up with two natural strategies for the Column player and we can think of \( \theta \) as the persistence of strategies. The purpose of this experiment is to elicit information about how humans persist in strategy choice, and we therefore investigate how the plays of the Row player can be used to obtain an estimate of \( \theta \) that is as precise as possible.

To cast this into our usual setting we think of \( S_t \) being the unobserved underlying Markov Chain, \( U_t \) as the control and \( Y_t \) as the observed process. Since the transition probabilities from \( S_t \) depend on \( U_{t-1} \) (a part of the history at time \( t-1 \)) we augment the state space to include \( U_{t-1} \), i.e. \( R_t = (S_t, U_{t-1}) \) will be our underlying Markov Chain. At this point we could run the dynamic programs for both FOFI and POFI, but controls calculated that way will depend deterministically on the plays of the Column player. Seeing that realistically deterministic plays can often easily be countered in adversarial games, it is better to follow a strategy that includes some randomness in the plays. So we let \( W_t \in \{-1, 1\} \) be the strategy of the Row player in such a way that

\[
\begin{align*}
U_t &= 1 \quad \text{w.p. .8} \quad \text{if } W_t = 1, \\
U_t &= -1 \quad \text{w.p. .2} \quad \text{if } W_t = -1
\end{align*}
\]

These kind of changes are easily incorporated in the dynamic program for both POFI and FOFI, by adding an expectation over \( W_t \) at every step \( t \).

We set \( \theta = .7 \) and calculated the FOFI and POFI policies. The long run FOFI policy was to alternate between \( W_t = 1 \) and \( W_t = -1 \) at every step (no matter the state of \( R_t \)). Thus if \( W_t = 1 \) at time \( t \), we choose \( U_t = 1 \) with probability .8 and then at time \( t + 1 \) we will set \( W_{t+1} = -1 \). The long run POFI policy depended on \( Y_t, Y_{t-1}, U_{t-1}, U_{t-2} \) in a rather complicated way, see Table 4.

To compare the two policies we ran a simulation study with \( T = 500 \), and 500 simulations for both the POFI and the FOFI policy. Another 500 simulations were run where the plays (control) were chosen randomly. The parameter \( \theta \) was estimated using an EM algorithm. The results of this estimation under each policy are given in Table 5 where the POFI controls produce least variance and the most accurate estimates.
Table 4: Long term POFI control policy: What $U_t$ to choose given different combinations of $Y_t,Y_{t-1},U_{t-1},U_{t-2}$.

| $U_{t-1}$ | $Y_t = 1$ | $Y_t = -1$ | $U_{t-2} = 1$ | $Y_{t-1} = 1$ | $Y_{t-1} = -1$ | $U_{t-2} = -1$ | $Y_{t-1} = 1$ | $Y_{t-1} = -1$ |
|-----------|-----------|-----------|---------------|---------------|---------------|---------------|---------------|---------------|
| 1         | 1         | 1         | -1            | -1            | -1            | -1            | -1            | -1            |
| -1        | -1        | -1        | 1             | 1             | 1             | 1             | 1             | 1             |

Table 5: Simulation results for the adversarial game. The POFI control policy seems to offer the best controls for optimizing the estimation of $\theta$. FOFI performs similarly to a random policy.

|               | bias | st. dev. | MSE |
|---------------|------|----------|-----|
| FOFI          | 0.004| 0.261    | 0.068|
| POFI          | 0.016| 0.231    | 0.054|
| Random        | 0.013| 0.252    | 0.064|
4 Discretization methods

In order to apply the methods described above to dynamical systems, we need to approximate them by a suitable Partially Observed Markov Decision Process. We achieve this by discretizing time, state and observation spaces. In this paper, the continuous stochastic dynamical systems considered are of the form

\[ dx = f(x, \theta, u(t))dt + \Sigma_1^{1/2}dW \]

where \( \theta \) is the parameter of interest, to be estimated, \( u(t) \) is a control that can be chosen by user, \( x \) is the vector of state variables, \( f \) is a vector valued function and \( W \) a Wiener process. The dynamical system is approximated on a fine grid of times \( (t\delta)_{t=0,...,T} \) and we obtain a discrete-time model

\[ x_{t+1} = x_t + \delta f(x_t, \theta, u_t) + \sqrt{\delta} \epsilon_{1t} \]

where \( \epsilon_{1t} \sim N(0, \Sigma_1) \) are independent normal random variables. We assume the underlying state variables \( x_t \) are only observed partially or noisily.

\[ y_t = g(x_t) + \epsilon_{2t} \]

where \( \epsilon_{2t} \sim N(0, \Sigma_2) \).

In order to approximate this as a Markov Chain, the state space is discretized in each dimension and the model is then thought of as moving between the different boxes. The probability of moving from box to box is approximated using the normal p.d.f. at the midpoints of the boxes. In the examples in this paper, only equidistant discretization is considered, but this restriction can be readily removed. If we label the two midpoints as \( i_1 \) and \( i_2 \) and the area of the second box as \( A_x \) this probability is given as

\[ p(x_{t+1} = i_2 | x_t = i_1, u_t, \theta) \approx \exp \left( -\frac{1}{2}(i_2 - (i_1 + \delta f(i_1, \theta, u_t)))^T \Sigma_1^{-1}(i_2 - (i_1 + \delta f(i_1, \theta, u_t))) \right) \cdot A_x \]

where \( k \) is the dimension of \( x \). The probabilities are then normalized to make sure they sum to 1. If the controls \( u_t \) can be chosen on a continuous scale then this scale has to be discretized as well. \( (x_t, u_t) \) is then a Markov Decision Process, and one can run the FOFI dynamic program.

For the POFI dynamic program the observation space needs to be discretized as well. The probability of what observation box is observed depends
on in which box the underlying Markov Chain is in. If we label the midpoint of the underlying Markov chain midpoint as \( i \) and the midpoint of the observed process box midpoint as \( j \), and the area of the latter box as \( A_y \) this probability is given as

\[
p(y_t = j | x_t = i) \approx \frac{1}{(2\pi)^{k/2} \det(\Sigma_2)^{1/2}} \exp\left(-\frac{1}{2}(j - g(i))^T \Sigma_2^{-1} (j - g(i))\right) \cdot A_y
\]

These probabilities are also normalized to sum to 1. The process \( (x_t, y_t, u_t) \) is now a Partially Observed Markov Decision Process and one can run the POFI dynamic program.

### 4.1 Morris Lecar Model

The Morris Lecar Model \[10\] describes oscillatory electric behavior in a single neural cell, as regulated by flow of Potassium and Calcium ions across the cell membrane. These models are defined in terms of state variables \( v_t \) and \( n_t \) representing the voltage across the membrane and the flux of the Potassium channel respectively.

\[
C_m \frac{dv_t}{dt} = I_t - g_l \cdot (v_t - E_l) - g_K \cdot n_t \cdot (v_t - E_K) - g_{Ca} \cdot m_\infty(v_t) \cdot (v_t - E_{Ca})
\]

\[
\dot{n}_t = -\phi \cdot (n_t - n_\infty(v_t)) / \tau_n(v_t)
\]

where \( m_\infty(v) = \frac{1}{2} (1 + \tanh((v - v_1)/v_2)) \), \( \tau_n(v) = \text{sech}((v - v_3)/(2v_4)) \) and \( n_\infty(v) = \frac{1}{2} (1 + \tanh((v - v_3)/v_4)) \). We will write \( C_m \dot{v}_t = F_1(v_t, n_t) \) and \( \dot{n}_t = F_2(v_t, n_t) \) as shortcuts equations \[1\] and \[2\]. The voltage between cells depends on Potassium and Calcium concentrations, and on the amount of leakage. The further these factors are away from their equilibriums \( E_l, E_K, E_{Ca} \) the greater the rate of change in voltage. The multiplicative value \( n_t \) changes the conductance of the potassium channel and is modeled through the second differential equation in which \( n_t \) is driven towards a voltage-dependent equilibrium level defined by \( n_\infty(v_t) \) but converges to this at a much slower rate then the dynamics of \( v_t \). The neuron is stimulated by an external applied current, \( I_t \) (our control), and \( v_t \) is measured. Our goal is to maximize information about the parameters \( C_m, g_{Ca} \) and \( \phi \), considered separately.
We consider a stochastic version of this neural firing model, derived from [9], by adding $\sigma dW_1$ and $\tilde{\sigma} dW_2$ to equations (1) and (2) respectively, where $W_1$ and $W_2$ are independent Wiener processes. Stochastic models are important in this context in order to accommodate observable variation in the inter-spike interval where a deterministic model will require a fixed period; see [3], for example.

The first step is to discretize these equations with respect to time. We get that $v_t(t + dt) = v(t) + dt \cdot F_1(v(t), n(t))/C_m + \sigma \sqrt{dt} \cdot \varepsilon_1$ and $n_t(t + dt) = n(t) + dt \cdot F_2(v(t), n(t)) + \tilde{\sigma} \sqrt{dt} \cdot \varepsilon_2$ where $\varepsilon_1, \varepsilon_2 \sim N(0, 1)$.

We discretized $v_t$ onto the range $[-75, 45]$ and $n_t$ onto $[0, 1]$, after running a few trial versions of the model. Both ranges where discretized into 25 intervals. Only $v_t$ is measured and it is measured noisily, $y_t = v_t + \varepsilon_t$ where $\varepsilon_t \sim N(0, 1)$. The observation space was discretized to the same range as $v_t$ but into 20 intervals. These approximations give rise to a Partially Observed Markov Decision Process to which our methods can be applied. Both FOFI and POFI controls were calculated for this experiment, for the parameters $C_m, g_{Ca}$ and $\phi$, considered separately. The values for the parameters were set to be $C_m = 20, g_{Ca} = 4.4, g_l = 2.0, E_k = -84.0, E_l = -60, E_{Ca} = 120.0, \phi = .04, v_1 = -1.2, v_2 = 18.0, v_3 = 2.0, v_4 = 30.0, \sigma = \tilde{\sigma} = 1$ and $dt = 1$. The controls range was set to be $[-1.5, 6.0]$ and discretized to the set $I_t \in \{-1.5, 0.0, 1.5, 3.0, 4.5, 6.0\}$. An example of what the control policy looks like is given in Figure 1. A simulation study was run for each of the three parameters; the system was simulated within the discretized Markov Chain framework with 100 time steps and all schemes had 100 simulations. The parameter in question was estimated for each simulation using an EM algorithm. As a baseline comparison we also ran a simulation study using a fixed control ($I_t = 1.5$). The results are given in Table 6. The difference between POFI and FOFI turns out to be not very dramatic, likely due to the observations providing a great deal of information about the underlying state variables, which is when FOFI performs well.

5 Parameter dependence of dynamic program

In the discussion above we calculated the dynamic program assuming knowledge of the parameter $\theta$, the very thing we wish to estimate with maximal
Figure 1: Long term controls of FOFI and POFI for the parameter $g_{Ca}$. The FOFI plot gives the control to use, given a certain position in state space. The POFI control will depend on the last two observations and the last control, but fixing the last control as, for example, $I_{t-1} = 6$ one can plot which control to use given combinations of the last two observations.
parameter & bias & st. dev. & MSE \\
\hline
$C_m$ & FOFI & .4234 & 2.4722 & 6.2913 \\
 & POFI & .4129 & 2.4068 & 5.9632 \\
 & Fixed & .9098 & 3.4240 & 12.551 \\
\hline
$g_{Ca}$ & FOFI & .0613 & .3671 & .1385 \\
 & POFI & .0158 & .3706 & .1376 \\
 & Fixed & .0249 & .6193 & .3841 \\
\hline
$\phi$ & FOFI & .00485 & .01085 & .00014 \\
 & POFI & .00257 & .01037 & .00011 \\
 & Fixed & .01357 & .02643 & .00088 \\
\hline

Table 6: Simulation results for the Morris-Lecar model, consider the parameters $C_m, g_{Ca}, \phi$ separately. We see that the POFI and FOFI policies outperform the fixed policy $I_t = 1.5$ in all cases, and the POFI policy seems to perform slightly better than the FOFI policy for the three parameters considered.

precision. Since the dynamic programs we have considered are run before the experiment is started we generally won’t have data to estimate $\theta$. Additionally, for the FOFI simulations we have used $\theta$ directly to estimate $x_t$ within the filter to get the appropriate control, but this will not be possible in practice. There are a few ways of dealing with this.

Assuming some prior information one can use a prior for $\theta$ to run the dynamic program. To do this, we add one more expectation for $\theta$ at every time step $t$, and then maximize the expected Fisher Information to get the best control. This strategy was employed in Lin et al. [4].

The rather obvious deficiency here, for both POFI and FOFI, is that as the experiment runs, we get observations that can be used to improve our prior for $\theta$, and could be used to get better controls, if we could brake the experiment and rerun the dynamic program.

5.1 Online updating

In some systems the time spent in each state is very short, too short to perform many calculations, making it valuable to have a “look-up table” of controls. Here the POFI controls have an advantage over the FOFI controls, in the sense that they are of the “look-up” kind, as FOFI requires estimation.
of the underlying $x_t$ process, before the control can be looked up.

In other systems, there is time to do some calculations between transitions. Note, for example, that at time $t$ we have observed $y_0, \ldots, y_t$ and this will allow us to calculate a posterior distribution $\pi(\theta|y_{0:t}, u_{0:t-1})$ for our parameter of interest. This posterior could then be used to run the dynamic program again, as described above, from time $T - 1$ to time $t$. This can be quite time consuming if done at each time step $t$, so we propose a method that relies on the Value Iteration Algorithm.

5.1.1 Value Iteration Algorithm

A popular algorithm from the theory of Markov Decision processes is the Value Iteration Algorithm (VIA), see Puterman [7]. The theoretical motivation of VIA is similar to dynamic programming, but here the objective is to maximize an expected total reward $W_1$ that has a discounting factor $\lambda$, where $0 \leq \lambda < 1$, and the time horizon is assumed to be infinite;

$$W_1 = E \left[ \sum_{t=0}^{\infty} \lambda^{t-1} C(x_t, u_t) \right]$$

and $W_1$ is labeled as the expected total discounted reward. $W_1$ exists if $C$ is bounded, which is the case in the problems we consider. In Puterman [7] it is shown that an optimal control policy exists and it can be chosen to be time independent, i.e. to depend only on the state $x_t$, and not the time $t$. Moreover this optimal control can be approximated using the Value Iteration Algorithm described below. Our experimental setting is neither discounted nor has it an infinite time horizon, but Blackwell optimality guarantees that controls that maximize $W_1$ also maximize the expected average reward $W_2$ (or its lim sup if the limit doesn’t exist);

$$W_2 = \lim_{n \to \infty} \frac{1}{n} E \left[ \sum_{t=0}^{n} C(x_t, u_t) \right]$$

given that $\lambda$ is chosen close enough to one. A Blackwell optimal control policy exists if the state and action spaces are finite, which is the case in our setting. Maximizing $W_2$ effectively amounts to maximizing the average input of each observation in our Fisher Information; a reasonable strategy. How small $1 - \lambda$ needs to be is generally hard to determine, and choosing $\lambda$ too
high will cause the algorithm to converge slowly. See Puterman [7] chapter 10 for more on Blackwell optimality. In VIA we calculate
\[ v^{n+1}(x_t, \theta) = \max_u \left\{ E_{x_{t+1}}[C(x_t, u_t, \theta) + \lambda \cdot v^n(x_{t+1}, \theta)|x_t, u_t, \theta] \right\} \]
in a while-loop until \( v^n \) converges to some fixed point, within some tolerance. Convergence is guaranteed since each iteration of \( v^n \) is a contraction mapping.

Our aim with VIA is to maximize what we in the POFI case label, the average Partial Observation Fisher Information
\[
\lim_{n \to \infty} \frac{1}{n} \mathbb{E}_{\theta} \mathbb{E}_{y|\theta} \sum_{t=0}^{n} \left( \frac{\partial}{\partial \theta} \log p(y_{t+1}|z_t, u_t, \theta) \right)^2
\]
(or in the FOFI case, the average Full Observation Fisher Information) which is a reasonable quantity to maximize in order to obtain a time-invariant policy.

We propose running VIA at every time step \( t \), but to use the posterior for \( \theta \), \( \pi(\theta|y_0:t, u_0:t-1) \), which is conditioned on all the data observed so far, instead of using the prior for \( \theta \). This will give a control that maximizes the average Fisher Information, using all the parameter information that is available at time \( t \). Instead of starting VIA at each time \( t \) with \( v^1 = 0 \), considerable time can be saved by using the last value vector \( v^n \) from the previous run of VIA at time \( t-1 \). This is because the posterior for \( \theta \) often doesn’t change much between time steps, and the last \( v^n \) from time \( t-1 \) thus being relatively close to the fixed point at time \( t \).

Let \( v^n_t \) denote the value vector at time \( t \) at the \( n \)'th iteration of the \( t \)'th VIA and let \( \pi(\theta|y_0:t, u_0:t-1) \) denote the posterior for \( \theta \) given observations up till time \( t \). Also, to ease notation, let \( z_t = y_{t-m:t}, u_{t-m:t-1} \). The pseudocode for this modified VIA using POFI is:

Set \( v^0_1 = 0 \) and \( n = 0 \)

for \( t = 0 \rightarrow T \)
do

\( \forall z_t \) and calculate and store

\[ v^{n+1}_t(z_t) = \max_{u_t} \sum_{\theta} \sum_{y_{t+1}} \left( \left( \frac{\partial}{\partial \theta} \log p(y_{t+1}|z_t, u_t, \theta) \right)^2 + \lambda v^n_t(z_{t+1})p(y_{t+1}|z_t, u_t, \theta)\pi(\theta|y_0:t, u_0:t-1) \right) \]

endwhile

endfor
n = n + 1
\textbf{end while}

Set \( v_{t+1}^0 = v_t^n \)

Now let
\[
u_t(z_t) = \arg\max_{u_t} \sum_{\theta} \sum_{y_{t+1}} \left[ \left( \frac{\partial}{\partial \theta} \log p(y_{t+1}|z_t, u_t, \theta) \right)^2 + \lambda v_t^n(z_{t+1}) p(y_{t+1}|z_t, u_t, \theta) \pi(\theta|y_{0:t-1}) \right] \]

Use control \( u_t \), and observe \( y_{t+1} \) and then update the posterior for \( \theta \),
\[
\pi(\theta|y_{0:t+1}, u_{0:t}) = \frac{p(y_{t+1}|y_{0:t}, u_{0:t}, \theta) \pi(\theta|y_{0:t-1})}{\sum_{\theta} p(y_{t+1}|y_{0:t}, u_{0:t}, \theta) \pi(\theta|y_{0:t-1})}
\]

\textbf{end for}

Updating FOFI policies online using VIA can be done in a similar way. In the next example we compare using updated policies.

### 5.2 PCR model

Polymerase chain reaction is a well established method to copy and multiply DNA. We are interested in modeling the growth dynamics of DNA template \((x_t)\), for a fixed amount of substrate. The model we use is
\[
x_{t+1} = (1 - u_t)x_t + dt \frac{a(1 - u_t)x_t}{(b + (1 - u_t)x_t)^2} + \sqrt{dt} \cdot \varepsilon_1
\]
where \( \varepsilon_1 \sim N(0, \sigma_1^2) \). Here \( x_t \) is the amount of DNA template, \( a \) and \( b \) the parameters of the model and \( u_t \) the control, the percentage of template removed at each time point. We are interested in estimating the parameter \( b \), labeled the half-saturation constant. A good reference for PCR models is [2].

We measure the amount of DNA template at each time point, but with an error. Our observations are
\[
y_t = x_t + \varepsilon_2 \text{ where } \varepsilon_2 \sim N(0, \sigma_2^2)
\]
and thus we have a dynamical system which when discretized becomes a Partially Observed Markov Decision Process.
The range for $x_t$ was set to be $[0, 15]$ and then discretized into 200 intervals, and $y_t$ was discretized to the same range, but only into 50 intervals. The parameter values were set to be $a = 2.0$, $b = 4.2$, $\sigma_1 = \sigma_2 = 1$, $dt = 1$ and the possible values of the control $u_t \in \{0, .2, .4, .6, .8, 1\}$.

Still with the objective of maximizing Fisher Information, we more realistically assumed priors for the parameters of the system, as discussed above. We conducted a simulation study using controls based on these priors for both POFI and FOFI, and then compared their performance to controls that are updated online using VIA, also both for POFI and FOFI. As a baseline comparison we also ran simulations using fixed controls and simulations where the true parameter is used (unrealistically) to calculate the control policy via dynamic programming as in the previous examples. For fixed controls we report the simulation with the lowest MSE, which was when $u_t = .2$.

The range for $b$ was set to be $b \in [1.7, 8.0]$ and then we discretized that interval into 10 points $\{1.7, 2.4, 3.1, 3.8, 4.5, 5.2, 5.9, 6.6, 7.3, 8.0\}$. We then considered a uniform prior on these points and a prior that is somewhat inaccurate, and puts the weight .9 on the point 7.3 and gives the others equal weight. The discounting factor for VIA was set to be $\lambda = .9$.

Our simulation study had the time length $T = 200$ and there were 600 simulations for each case. The parameter $b$ was estimated using an EM algorithm. The simulation results are given in Table 7.

We note that when we calculate the controls prior to the experiment (No online updating), both the POFI and FOFI controls are significantly better than using a fixed control, and POFI seems to do better than FOFI when we use an uniform prior. Interestingly in the FOFI case, calculating the controls using the inaccurate prior does better then using the uniform prior, likely due to a reduction in prior variance, in spite of additional bias.

Accuracy increases in most cases when we allow for online updating using the VIA algorithm. Starting the VIA with an uniform prior does better than starting with the inaccurate one, which is probably due to the VIA having to spend more time “repairing” the prior. Also, we note that VIA controls with uniform prior have a similar performance to a control policy using the true (unknown) parameter.

Additionally, in Figure 2 we see that using the previous final value vector as the starting value vector of VIA when going from time point $t$ to $t + 1$, does save considerable time, and more so as $t$ grows and the posterior for the parameter starts to change less.
Table 7: Simulation results for the PCR Model using two kinds of priors, POFI and FOFI and with and without VIA.

|                      | bias | st. dev. | MSE     |
|----------------------|------|----------|---------|
| uniform prior, without VIA |      |          |         |
| FOFI                 | 0.1059 | 0.6598   | 0.4465  |
| POFI                 | 0.0053 | 0.6189   | 0.3831  |
|                      |      |          |         |
| inaccurate prior, without VIA |      |          |         |
| FOFI                 | 0.0755 | 0.6374   | 0.4120  |
| POFI                 | 0.0516 | 0.7051   | 0.4998  |
|                      |      |          |         |
| fixed control (\(u_t = .2\) for all \(t\)) |      |          |         |
| fixed               | .1264 | .7466    | .5734   |
|                      |      |          |         |
| uniform prior, with VIA |      |          |         |
| FOFI                 | 0.0388 | 0.6180   | 0.3834  |
| POFI                 | 0.0766 | 0.5999   | 0.3658  |
|                      |      |          |         |
| inaccurate prior, with VIA |      |          |         |
| FOFI                 | 0.0713 | 0.6787   | 0.4657  |
| POFI                 | 0.0954 | 0.6750   | 0.4648  |
|                      |      |          |         |
| True parameter, without VIA |      |          |         |
| FOFI                 | 0.0659 | 0.6235   | 0.3932  |
| POFI                 | 0.0323 | 0.6249   | 0.3916  |

Figure 2: Running time of VIA at each time step \(t\), for POFI using a uniform prior for the PCR model.
6 Discussion

In this paper we compared two possible ways to conduct experimental design in parametric POMDP’s, based on using dynamic programming to maximize either the Partial Observation Fisher Information or the Full Observation Fisher Information. Settings can arise where controls chosen by FOFI are not optimal, due to focusing on the underlying process rather than the observed process, and in these cases controls chosen with POFI often perform better, as in the six state example and the adversarial game. In some of the examples analyzed they performed similarly.

In recent years, there has been growing interest in statistical procedures within dynamical systems, such as parameter estimation and hypothesis testing, and many of these procedures could be performed more efficiently given good experimental design. In this paper we fully discretized the state and observational spaces to transform dynamical systems with stochastic errors into partially observed Markov decision processes, allowing us to use the methods developed for POMDP’s to our advantage.

We also noted how the problem of parameter dependence can be overcome by averaging over a prior. Additionally given that there is enough time between consecutive time steps, we showed how the controls can be efficiently updated online using observations gathered so far, by using a variant of the Value Iteration Algorithm. This was demonstrated in the PCR example.

Finding controls that maximize information about parameters is a computationally challenging task. We have successfully demonstrated techniques for up to two dimensional systems, for a one dimensional parameter. Adding dimensions in state, parameter or observation space quickly make the methods considered computationally intractable. Considering a longer lag of past observations for POFI might also increase accuracy, but again at the cost of computation time. In order to extend these methods to higher dimensions one could possibly use techniques of approximate dynamic programming, see Powell [6].

References

[1] O. Cappe, Moulines E., and Ryden T. *Inference in Hidden Markov Models*. Springer, 2005.
[2] P. Haccou, P. Jagers, and V.A. Vatutin. Branching processes: Variation, growth, and extinction of populations, volume 5. Cambridge Univ Pr, 2005.

[3] G. Hooker. Forcing function diagnostics for nonlinear dynamics. Biometrics, 65:613–620, 2009.

[4] K. K. Lin, G. Hooker, and B. Rogers. Control theory and experimental design in diffusion processes. Unpublished, Department of Biological Statistics and Computational Biology, Cornell University. 2012.

[5] G.E. Monahan. A survey of partially observable markov decision processes: Theory, models, and algorithms. Management Science, 28(1):1–16, 1982.

[6] W. Powell. Approximate Dynamic Programming: solving the curses of dimensionality. Wiley, 2007.

[7] M.L. Puterman. Markov Decision Processes - Discrete Stochastic Dynamic Programming. Wiley, Hoboken, NJ, 2005.

[8] J. Shachat, J. T. Swarthouty, and L. Wei. Man versus nash: An experiment on the self-enforcing nature of mixed strategy equilibrium. Unpublished, Wang Yanan Institute for Studies in Economics, Xiamen University. 2011.

[9] Gregory Smith. Modeling the stochastic gating of ion channels. In Computational Cell Biology, volume 20 (II) of Interdisciplinary Applied Mathematics. 2002.

[10] D. Terman and B. Ermentrout. Mathematical Foundations of Neuroscience. Springer, 2010.
A Appendix 1

A.1 Pseudocode for POFI and FOFI

A.1.1 Dynamic program for approximated POFI

We maximize

\[ FI_{approx} = E \sum_{t=0}^{T-1} \left( \frac{\partial}{\partial \theta} \log p(y_{t+1}|y_{t-m:t}, u_{t-m:t}, \nu_{t-m}, \theta) \right)^2 \]

with reward function \( C(y_{t-m:t}, u_{t-m:t}, \theta) = \left( \frac{\partial}{\partial \theta} \log p(y_{t+1}|y_{t-m:t}, u_{t-m:t}, \theta) \right)^2 \)

and Fisher Information To Go

\[ FI_{t,m}(y_{t-m:t}, u_{t-m:t-1}, \theta) = \max \{ E_{y_{t+1}}[C + FI_{t+1,m}(y_{t-m:t}, u_{t-m:t}, \theta)] \} \]

The pseudocode for this dynamic program is:

\[
\begin{align*}
FI_{T,m} &= 0 \\
\text{for } t &= (T-1) \rightarrow 0 \text{ do} \\
\forall y_{t-m:t}, u_{t-m:t-1} \text{ and calculate and store} \\
FI_{t,m}(y_{t-m:t}, u_{t-m:t-1}, \theta) &= \max_{u_t} \left\{ E_{y_{t+1}}[C + FI_{t+1,m}(y_{t-m:t}, u_{t-m:t}, \theta)] \right\} \\
u^*_t(y_{t-m:t}, u_{t-m:t-1}, \theta) &= \arg \max_{u_t} \left\{ E_{y_{t+1}}[C + FI_{t+1,m}(y_{t-m:t}, u_{t-m:t}, \theta)] \right\} \\
\text{end for}
\end{align*}
\]

A.1.2 Dynamic program for FOFI

We maximize

\[ FI = E \sum_{t=0}^{T-1} \left( \frac{\partial}{\partial \theta} \log p(x_{t+1}|x_t, u_t, \theta) \right)^2 \]

by setting the reward function as \( C'(x_t, u_t, \theta) = \left( \frac{\partial}{\partial \theta} \log p(x_{t+1}|x_t, u_t, \theta) \right)^2 \). The pseudocode for this dynamic program is:

\[
\begin{align*}
FI_{T} &= 0 \\
\text{for } t &= (T-1) \rightarrow 0 \text{ do} \\
\forall x_t \text{ and calculate and store} \\
FI_{t}(x_t, \theta) &= \max_{u_t} \left\{ E_{x_{t+1}}[C(x_t, u_t, \theta) + FI_{t+1}(x_{t+1}, \theta)|x_t, u_t, \theta)] \right\} \\
u^*_t(x_t, \theta) &= \arg \max_{u_t} \left\{ E_{x_{t+1}}[C(x_t, u_t, \theta) + FI_{t+1}(x_{t+1}, \theta)|x_t, u_t, \theta)] \right\} \\
\text{end for}
\end{align*}
\]
A.2 Expressing Fisher Information

In this section we find useful expressions for the Fisher Information of an experiment, that are needed to derive convergence arguments and set up dynamical programs. We use the short hand notation

\[ h_k(\theta) = \log p(y_k|y_{0:k-1}, u_{0:k-1}, x_0 = x) \]

where the dependence on \( X_0 = x \) is suppressed.

For data \( Y_0, \ldots, Y_T \) the Fisher Information for \( \theta \) can be expressed in one or two derivatives

\[ FI = E \left[ T-1 \sum_{t=0}^{T-1} \ddot{h}_{t+1} \right] = E \left[ \left( \sum_{t=0}^{T-1} \dot{h}_{t+1} \right)^2 \right] \]

and similarly the Fisher Information to Go

\[ FI_k = E \left[ \sum_{t=k}^{T-1} -\ddot{h}_{t+1} \mid y_{0:k}, u_{0:k-1} \right] = E \left[ \left( \sum_{t=k}^{T-1} \dot{h}_{t+1} \right)^2 \right] \]

where the equality is justified by both quantities being the Fisher Information for the same observations. We set \( FI_0 = FI \).

The Fisher Information to Go can be calculated recursively (in both one or two derivatives):

**Lemma 1.**

\[ FI_k = E \left[ -\ddot{h}_{k+1} + FI_{k+1} \mid y_{0:k}, u_{0:k-1} \right] = E \left[ (\dot{h}_{k+1})^2 + FI_{k+1} \mid y_{0:k}, u_{0:k-1} \right] \]

**Proof.** In the case of using two derivatives this follows from iterated expectation. In one derivative we have

\[ FI_k = E \left[ \left( \sum_{t=k}^{T-1} \dot{h}_{t+1} \right)^2 \right] \]

\[ = E \left[ (\dot{h}_{k+1})^2 + \left( \sum_{t=k+1}^{T-1} \dot{h}_{t+1} \right)^2 + 2 (\dot{h}_{k+1}) \left( \sum_{t=k+1}^{T-1} \dot{h}_{t+1} \right) \mid y_{0:k}, u_{0:k-1} \right] \]

27
The cross term is
\[
E \left[ 2 \left( \hat{h}_{k+1} \right) \left( \sum_{t=k+1}^{T-1} \hat{h}_{t+1} \right) \right| y_{0:k}, u_{0:k-1}]
\]
\[
= E \left[ 2 \left( \hat{h}_{k+1} \right) \left( \sum_{t=k+1}^{T-1} \hat{h}_{t+1} \right) \right| y_{0:k+1}, u_{0:k} \right] y_{0:k}, u_{0:k-1}
\]
\[
= E \left[ 2 \left( \hat{h}_{k+1} \right) E \left[ \left( \sum_{t=k+1}^{T-1} \hat{h}_{t+1} \right) y_{0:k+1}, u_{0:k} \right] \right| y_{0:k}, u_{0:k-1}
\]
\[
= E \left[ 2 \left( \hat{h}_{k+1} \right) \cdot 0 \right| y_{0:k}, u_{0:k-1} = 0
\]

Thus

\[
FI_k = E \left[ \left( \hat{h}_{k+1} \right)^2 + \left( \sum_{t=k+1}^{T-1} \hat{h}_{t+1} \right)^2 \right| y_{0:k}, u_{0:k-1}
\]
\[
= E \left[ \left( \hat{h}_{k+1} \right)^2 \right] + E \left[ \left( \sum_{t=k+1}^{T-1} \hat{h}_{t+1} \right)^2 \right| y_{0:k}, u_{0:k-1}
\]
\[
= E \left[ \left( \hat{h}_{k+1} \right)^2 + FI_{k+1} \right| y_{0:k}, u_{0:k-1}
\]

\[
\square
\]

**Corollary 1.**

\[
FI = E \left[ \sum_{t=0}^{T-1} \left( \hat{h}_{t+1} \right)^2 \right]
\]

*and similarly*

\[
FI_k = E \left[ \sum_{t=k}^{T-1} \left( \hat{h}_{t+1} \right)^2 \right| y_{0:k}, u_{0:k-1}
\]

**Proof.** This follows from using induction and lemma 1. \(\square\)

28
A.2.1 Approximating the Fisher Information to Go

Running an exact dynamic program, with $F_I_k$ as the value function, is not feasible because of the curse of dimensionality, leading us to certain approximations. We set

$$h_{k,m,\nu_m}(\theta) = \begin{cases} 
\log p(y_k|y_{m:k-1}, u_{m:k-1}, \nu_m) & \text{if } m \geq 0 \\
\log p(y_k|y_{0:k-1}, u_{0:k-1}, \nu_0) & \text{if } m < 0
\end{cases}$$

where $\nu_m$ is the assumed distribution of $x_m$ and we will consider it to be fixed and known. Allowing $m$ to be negative will ease notation when $t - m < 0$. We now set

$$F_I_{k,m} = E \left[ \sum_{t=k}^{T-1} \left( \dot{h}_{t+1,t-m,\nu_t-m} \right)^2 | y_{k-m:k}, u_{k-m:k-1} \right]$$

$$= E \left[ \sum_{t=k}^{T-1} -\ddot{h}_{t+1,t-m,\nu_t-m} | y_{k-m:k}, u_{k-m:k-1} \right]$$

and denote it the approximated Fisher Information to Go. That the formulation in one derivative is equal to the one in two derivatives follows from the individual parts of each sum having a Fisher Information interpretation.

A.3 Mixing conditions

Cappe et al. establish forgetting properties of the filter by assuming mixing conditions for Hidden Markov Models. We use the same conditions, slightly modified to allow for controls, for POMDPs;

Assumption 1. Modified Strong Mixing Conditions. For each control $u$ there exist a transition kernel $K^u: Y \to X$ and measurable functions $\varsigma^-$ and $\varsigma^+$ from $Y$ to $(0, \infty)$ such that for any $A \in \mathcal{X}$, $y \in Y$ and $x \in X$,

$$\varsigma^-(y)K^u(y, A) \leq \sum_{x' \in A} p(y_{t+1} = y|x_{t+1} = x')p(x_{t+1} = x'|x_t = x) \leq \varsigma^+(y)K^u(y, A)$$

Cappe et al.’s discussion on what models satisfy these conditions applies analogously to POMDP’s. Given these conditions we can prove the following, which is a modification of Lemma 12.5.3 in [1];
Theorem 3. Assume the strong mixing conditions in Assumption 1. Then

\[
(E(\hat{h}_{k,0,\nu_0}(\theta) - \hat{h}_{k,m,\nu_m}(\theta))^2)^{1/2} \leq 8 \sup_{x,x' \in X, u \in U, y \in Y} ||\phi_\theta(x, x', y, u)|| \frac{\rho^{(k-m)/2-1}}{1 - \rho}
\]

where \(\phi(x, x', u, y) = \frac{\partial}{\partial \theta} \log(p(x_{t+1} = x' | x_t = x, u_t = u, \theta)p(y_{t+1} = y' | x_{t+1} = x', \theta))\) and \(\rho = \max_{y \in Y} 1 - \frac{c^- (y)}{c^+ (y)}\)

A proof is provided in Appendix 2.

A.4 Bounds on Fisher Information

In this section we show that the approximated Fisher Information approaches the true Fisher Information exponentially as one conditions on more and more observations, while using the same controls.

By Corollary 1 the true Fisher Information is

\[
FI(\theta, u_{0:T-1}) = E \sum_{k=0}^{T-1} (\dot{h}_{k+1,0,\nu_0}(\theta))^2
\]

but since that is hard to optimize we consider

\[
FI_{approx} = FI_{0,m}(\theta, u_{0:T-1}) = E \sum_{k=0}^{T-1} (\dot{h}_{k+1,k-m,\nu_{k-m}}(\theta))^2
\]

see definitions for \(\dot{h}\) above. Here we use Fisher Information in one derivative, but as noted above it is equivalent to using the formulation in two derivatives. Also note that where \(k - m < 0\) we just set it to 0 and use the initial distribution of \(x_0\).

Lemma 2. Assume the mixing conditions in Assumption 1 hold. Then

\[
\left( E(\dot{h}_{k+1,0,\nu_0} + \dot{h}_{k+1,k-m,\nu_{k-m}})^2 \right)^{1/2} \leq 16 \sup_{x,x' \in X, u \in U, y \in Y} |\phi_\theta(x, x', y, u)| \frac{\rho^{1/2}}{1 - \rho} + 2 \sup_{u_0} \left( E(\dot{h}_{1,0,\nu_0})^2 \right)^{1/2}
\]
Proof. Set

\[ A(m') = \sup_{u_1, \ldots, u_{m'-1}} \left( E(\dot{h}_{m',0,\nu_0})^2 \right)^{1/2} \]

which sets an upper bound on the length of \( \dot{h}_m \). Note that \( A(m') \) also bounds \( (E(\dot{h}_{k+1,k-m,\nu_{k-m}})^2)^{1/2} \) since \( \nu_{k-m} = \nu_0 \). Now

\[
\left( E(\dot{h}_{k+1,0,\nu_0} + \dot{h}_{k+1,k-m,\nu_{k-m}})^2 \right)^{1/2} \\
\leq \left( E(\dot{h}_{k+1,0,\nu_0} - \dot{h}_{k+1,k-m',\nu_{k-m'}})^2 \right)^{1/2} + \left( E(\dot{h}_{k+1,k-m,\nu_{k-m}} - \dot{h}_{k+1,k-m',\nu_{k-m'}})^2 \right)^{1/2} \\
+ 2 \left( E(\dot{h}_{k+1,k-m',\nu_{k-m'}})^2 \right)^{1/2} \\
\leq 16 \sup \phi_{\theta} \frac{\rho^{(1+\min(m,m'))/2}}{1-\rho} + 2A(m' + 1)
\]

using Theorem \[5\]. Setting \( m' = 0 \) gives the result, although that might not be the best bound. \(\square\)

We now restate Theorem \[1\] from section 2.1.

**Theorem 1.** Assume the conditions in Assumption \[4\] hold. Then, for \( m < T \) and any control policy, we have

\[
|FI - FI_{0,m}| \leq c_1 (T - 1 - m) \rho^{m/2}
\]

where \( c_1 = 8M(\theta) \sup_{x,x',y} \phi_{\theta}(x, x', y, u) \frac{1}{\rho^{1/2}(1-\rho)} \) and \( M(\theta) \) is the bound from lemma \[2\]. \( M(\theta) = 16 \sup \phi_{\theta} \frac{\rho^{1/2}}{1-\rho} + 2 \sup_{u_0} \left( E(\dot{h}_{1,0,\nu_0})^2 \right)^{1/2} \).

Proof.

\[
|FI - FI_{0,m}| = \left| E \sum_{k=0}^{T-1} \dot{h}_{k+1,0,\nu_0}(\theta)^2 - E \sum_{k=0}^{T-1} \dot{h}_{k+1,k-m,\nu_{k-m}}(\theta)^2 \right| \\
= \left| \sum_{k=m+1}^{T-1} E(\dot{h}_{k+1,0,\nu_0} - \dot{h}_{k+1,k-m,\nu_{k-m}}) \right| \\
\leq \sum_{k=m+1}^{T-1} |E(\dot{h}_{k+1,0,\nu_0} - \dot{h}_{k+1,k-m,\nu_{k-m}}) \cdot (\dot{h}_{k,0,\nu_0} + \dot{h}_{k,k-m,\nu_{k-m}})|
\]

31
and by Cauchy Schwarz
\[
\leq \sum_{k=m+1}^{T-1} \left( E \left| \dot{h}_{k+1,0,\nu_0} - \dot{h}_{k+1,k-m,\nu_{k-m}} \right|^2 \right)^{1/2} \cdot \left( E \left| \dot{h}_{k,0,\nu_0} + \dot{h}_{k,k-m,\nu_{k-m}} \right|^2 \right)^{1/2}
\]

For the first parenthesis we use Theorem 5 to get
\[
\left( E \left| \dot{h}_{k+1,0,\nu_0} - \dot{h}_{k+1,k-m,\nu_{k-m}} \right|^2 \right)^{1/2} \leq 8 \sup_{x,x \in X,u \in U,y \in Y} |\phi_\theta(x,x,y,u)| \frac{\rho^{(m+1)/2-1}}{1 - \rho}
\]
and the second one is bounded by the Lemma 2. We get
\[
|FI - FI_{0,m}| \leq 8M(\theta) \sup_{x,x \in X,u \in U,y \in Y} |\phi_\theta(x,x,y,u)| \sum_{k=m+1}^{T-1} \frac{\rho^{(m+1)/2-1}}{1 - \rho}
\]
\[
= 8M(\theta)(T - 1 - m) \sup_{x,x \in X,u \in U,y \in Y} |\phi_\theta(x,x,y,u)| \frac{\rho^{(m+1)/2-1}}{1 - \rho}
\]

Exactly the same arguments can be used to show that the approximated Fisher Information to Go $FI_{k,m}$ approaches the true Fisher Information to Go as $m$ increases.

### A.5 Best Possible Fisher Information

A related problem we are interested in is how well controls that consider the last say $m$ observations do in comparison with controls that consider all past observations. That is, we want a bound on the best possible Fisher Information given controls that consider all past observations, compared with the best possible Fisher Information given controls that only consider the last $m$ observations.

We first establish a baseline difference between two parts of the Fisher Information.

**Lemma 3.** Assume the conditions in Assumption 1 hold. Then, for any control policy and any $k$, we have
\[
|E(\dot{h}_{k+1,0,\nu_0}^2 - \dot{h}_{k+1,k-m,\nu_{k-m}}^2)| \leq 8M(\theta) \sup_{x,x' \in X,u \in U,y \in Y} |\phi_\theta(x,x',y,u)| \frac{\rho^{(m+1)/2-1}}{1 - \rho}
\]
where $M(\theta) = 16 \sup |\phi_\theta|^{\theta^{1/2}} + 2 \sup_{u_0} \left( E(\hat{h}_{1,0,v_0})^2 \right)^{1/2}$ is the bound from lemma 2.

Proof. Follows from Lemma 2 and Theorem 5 as in the proof of Theorem 1. 

As above, we assume that the Fisher Information to Go;

$$FI_k = E \left[ (\hat{h}_{k+1,0,v_0})^2 + FI_{k+1} \Bigg| y_{0:k}, u_{0:k-1} \right]$$

is approximated by

$$FI_{k,m} = E \left[ (\hat{h}_{k+1,k-m,v_{k-m}})^2 + FI_{k+1,m} \Bigg| y_{k-m:k}, u_{k-m:k-1} \right]$$

Given that our controls are obtained by dynamic programming, we have that the optimal control at time $t$ is dependent on the optimal control obtained at time $t + 1$. Let $u_1^*, \ldots, u_{T-1}^*$ denote the set of optimal controls obtained in this manner, i.e. $u_k^*, \ldots, u_T^*$ maximize $FI_k$.

As argued these controls are hard to calculate and thus we resort to approximate optimal controls, here denoted $u_0^*, m, \ldots, u_{T-1}^*, m$, where $u_k^*, m, \ldots, u_T^*, m$ maximize $FI_{k,m}$.

We now restate Theorem 2 from section 2.3 on the loss of using approximate controls instead of exact ones in Fisher Information, in an experiment of length $T$.

**Theorem 2.** Given that the mixing conditions in Assumption hold we have

$$0 \leq FI(u_0^*, \ldots, u_{T-1}^*) - FI(u_0^*, m, \ldots, u_{T-1}^*, m) \leq c_2 T(T + 1) \rho^{m/2}$$

where $c_2 = 8M(\theta) \sup |\phi_\theta(x, x', y, u)| \frac{1}{\rho^{m/2}(1-\rho)}$, and $M(\theta)$ is the bound from lemma 2.

Proof. We analyze the difference by bounding errors in each step of the dynamic program inductively, starting at time $t = T - 1$ and going backwards. Set $e = 8M(\theta) \sup |\phi_\theta(x, x', y, u)| \frac{\rho^{(m+1)/2-1}}{1-\rho}$.

We find that

$$0 \leq FI_{T-1}(u_{T-1}^*) - FI_{T-1}(u_{T-1}, m)$$

$$\leq FI_{T-1}(u_{T-1}^*) - FI_{T-1}(u_{T-1}, m) + (FI_{T-1,m}(u_{T-1}, m) - FI_{T-1,m}(u_{T-1}))$$

33
so far only using that $u^*_T$ maximizes $F I_{T-1}$ and $u^*_{T-1,m}$ maximizes $F I_{T-1,m}$. 

\[
|F I_{T-1}(u^*_T) - F I_{T-1,m}(u^*_T)| + |F I_{T-1}(u^*_{T-1,m}) - F I_{T-1,m}(u^*_{T-1,m})| 
\leq 2\varepsilon
\]

by lemma 3.

We now inductively assume

\[
|F I_{T-s}(u^*_{T-s:T-1}) - F I_{T-s}(u^*_{T-s:T-1,m})| \leq s(s + 1)\varepsilon
\]

where $u^*_{T-s:T-1,m} = u^*_{T-s, m}, \ldots, u^*_{T-1,m}$, and then get

\[
|F I_{T-s}(u^*_{T-s:T-1}) - F I_{T-s,m}(u^*_{T-s:T-1,m})| \leq |F I_{T-s}(u^*_{T-s:T-1}) - F I_{T-s}(u^*_{T-s:T-1,m})| + |F I_{T-s}(u^*_{T-s:T-1,m}) - F I_{T-s,m}(u^*_{T-s:T-1,m})| 
\leq s(s + 1)\varepsilon + s\varepsilon = s(s + 2)\varepsilon
\]

(1)

Now moving from $s$ to $s + 1$ we have

\[
F I_{T-(s+1),m}(u^*_{T-(s+1):T-1,m}) \geq F I_{T-(s+1),m}(u^*_{T-(s+1):T-1,m})
\]

since $u^*_{T-(s+1):T-1,m}$ are the controls that maximize $F I_{T-(s+1),m}$. By adding and subtracting the same quantity we get the following equivalent inequality

\[
(F I_{T-(s+1),m}(u^*_{T-(s+1):T-1,m}) - F I_{T-(s+1)}(u^*_{T-(s+1):T-1,m})) = (2)
\]

\[
- (F I_{T-(s+1),m}(u^*_{T-(s+1)}, u^*_{T-s:T-1,m}) - F I_{T-(s+1)}(u^*_{T-(s+1):T-1})) = (3)
\]

\[
\geq F I_{T-(s+1)}(u^*_{T-(s+1):T-1}) - F I_{T-(s+1)}(u^*_{T-(s+1):T-1,m}) \geq 0
\]

Line (2) is bounded by $(s + 1)\varepsilon$ by lemma 3 and line (3) by $\varepsilon + s(s + 2)\varepsilon$ using (1) and lemma 3. Therefore

\[
|F I_{T-(s+1)}(u^*_{T-(s+1):T-1}) - F I_{T-(s+1)}(u^*_{T-(s+1):T-1,m})| \leq (s + 1)\varepsilon + \varepsilon + s(s + 2)\varepsilon = (s + 1)(s + 2)\varepsilon
\]

and for the whole experiment we find

\[
|F I(u^*_{0:T-1}) - F I(u^*_{0:T-1,m})| \leq T(T + 1)\varepsilon
\]
Appendix 2

B.1 The different computational complexity of POFI and FOFI

We analyze the computational complexities of FOFI and POFI. The computations required can be split into computations done prior to the experiment and computations that are required while running the experiment. A direct comparison is not completely fair since FOFI requires computations at runtime while POFI does not as discussed below.

B.1.1 FOFI

Prior to the experiment we use a dynamic program that provides us with a control-policy that maximizes

\[ E \sum_{t=0}^{T-1} \left( \frac{\partial}{\partial \theta} \log p(x_{t+1}|x_t, u_t, \theta) \right)^2, \]

i.e. the Full Observation Fisher Information.

We assume that the transition probability matrix \( p(x_{t+1}|x_t, u_t, \theta) \) is given. Calculating \( \left( \frac{\partial}{\partial \theta} \log p(x_{t+1}|x_t, u_t, \theta) \right)^2 \) is negligible compared to the calculations required for the dynamic program; If we set

\[ g_t(x_t, x_{t+1}, u_t, \theta) = \left( \frac{\partial}{\partial \theta} \log p(x_{t+1}|x_t, u_t, \theta) \right)^2 \]

then for a given time \( t \) in the dynamic program we need to maximize

\[ E [g_t(x_t, x_{t+1}, u_t, \theta) + V_{t+1}(x_{t+1}, \theta)|x_t] \]

over \( u_t \in U \) for each \( x_t \in \mathcal{X} \), where \( V_{t+1} \) is the value function from the previous step \( t + 1 \). This calculation requires adding \( g_t \) and \( V_{t+1} \) which are two \( K \times 2 \times l \) tensors with cost \( K^2l \). Next we need a dot product between \( g_t + V_{t+1} \) and \( p(x_{t+1}|x_t, u_t) \) over the \( x_{t+1} \) dimension which has cost \( O(K^2l) \). Finally maximizing over \( u_t \) for each \( x_t \) has cost \( O(Kl) \). Thus each step \( t \) has cost \( O(K^2l) \) and the dynamic program in total has cost \( O(TK^2l) \).

In runtime a filter is required to estimate the state \( x_t \). The filter for time \( t + 1 \) can be calculated via the following recursive formula

\[ p(x_{t+1}|y_{0:t+1}, u_{0:t}) \propto \sum_{x_t} p(y_{t+1}|x_{t+1})p(x_{t+1}|x_t, u_t)p(x_t|y_{0:t}, u_{0:t-1}) \]
and then normalizing. This requires $2K$ dot products of vectors of length $K$, with cost $O(K^2)$ and the normalization has cost $O(K)$. Thus we have $O(K^2)$ computations at each time step $t$ during runtime.

B.1.2 POFI

Here the dynamic program maximizes the approximated Partial observation Fisher Information,

$$E \left[ \sum_{t=0}^{T-1} \left( \frac{\partial}{\partial \theta} \log p(y_{t+1}|y_{0:t}, u_{0:t}, x_0, \theta) \right)^2 \right]$$

First we note that $p(y_{t+1}|y_{m:t}, u_{m:t}, \theta)$ is a $L^{x+m+2} \times L^{x+1}$ tensor, and it can be calculated using Bayes rule at the cost $O(K^2L^{m+2l}m+1)$. Calculating $\frac{\partial}{\partial \theta} \log p(y_{t+1}|y_{m:t}, u_{m:t})$ can also be done at the cost $O(K^2L^{m+2l}m+1)$, but can also be effectively approximated using the finite difference approximation to the derivative.

The cost analysis of the POFI dynamic program is just like the analysis of FOFI. At a given time $t$ adding $g_t$ and $V_{t+1}$ has cost $O(L^{m+2l}m+1)$, the dot product between $g_t + V_{t+1}$ and $p(y_{t+1}|y_{m:t}, u_{m:t})$ has cost $O(L^{m+2l}m+1)$ and the maximization has cost $O(L^{m+1}l^{m+1})$.

The dynamic program thus has cost $O( TL^{m+2l}m+1)$, which we kept from growing to large by choosing $L$ significantly lower than $K$ and $m = 1$.

B.2 Modified HMM theory

This section is devoted to expanding Hidden Markov Model Theory to Partially Observed Markov Decision Processes. We base it completely on Cappe et al. \[1\] and use their notation, only changing what is necessary. The purpose is to prove Theorem 3 in Appendix A.3, which is a modified version of Lemma 12.5.3 in \[1\]. In most cases the changes will amount to adding controls and seeing that the theory follows through, although the proof of Theorem 3 has more substantial changes.

B.2.1 Setup

Let $(X, \mathcal{X})$ and $(Y, \mathcal{Y})$ be the state space and the observations space respectively. Let

$$Q^u(x, A) = \int_A q^u(x, x') dx', \ A \in \mathcal{X}, u \in \mathcal{U}$$
be a transition kernel for our state space, where \( u \) is a control, and \( \mathcal{U} \) is finite. Also let
\[
G(x, A) = \int_A g(x, y) dy, \ A \in \mathcal{Y}
\]
be the transition kernel for moving from the state space to the observation space.

We generally assume that the Markov Chain is initialized with distribution \( \nu \), and then runs for \( n \) steps \( x_{0:n} = x_0, \ldots, x_n \) and that \( n - 1 \) decisions are made on what controls \( u \) to use. This results in \( n \) observations \( y_{0:n} = y_0, \ldots, y_n \) and \( n - 1 \) control \( u_{0:n-1} = u_0, \ldots, u_{n-1} \).

**B.2.2 Hidden Markov Model theory**

**Definition 1** (Definition 3.1.6 in [1]). Conditional on \( y_{0:k} \) and \( u_{0:k-1} \) we define the forward variable
\[
\alpha_{\nu,k}(y_{0:k}, u_{0:k-1}, f) = \ldots \int f(x_k) \nu(dx_0) g(x_0, y_0) \prod_{l=1}^{k} Q^{u_{l-1}}(x_{l-1}, dx_l) g(x_l, y_l)
\]
and conditional on \( y_{k+1:n} \) and \( u_{k:n} \) we define the backward variable
\[
\beta_{k|n}(y_{k+1:n}, u_{k:n}, x_k) = \ldots \int Q^{u_k}(x, dx_{k+1}) g(x_{k+1}, y_{k+1}) \prod_{l=k+2}^{n} Q^{u_{l-1}}(x_{l-1}, dx_l) g(x_l, y_l)
\]

As in the classical case these satisfy recursion formulas
\[
\alpha_{\nu,k}(y_{0:k}, u_{0:k-1}, f) = \int f(x_k) \int \alpha_{\nu,k-1}(y_{0:k-1}, u_{0:k-2}, dx_{k-1}) Q^{u_{k-1}}(x_{k-1}, dx_k) g(x_k, y_k)
\]
with initial condition
\[
\alpha_{\nu,0}(f) = \int f(x_0) g(x_0, y_0) \nu(dx_0)
\]
and similarly
\[
\beta_{k|n}(y_{k+1:n}, u_{k:n}, x_k) = \int Q^{u_k}(x, dx_{k+1}) g(x_{k+1}, y_{k+1}) \beta_{k+1|n}(y_{k+2:n}, u_{k+1:n}, x_{k+1})
\]

A standard result in HMM theory is that conditional on the observations \( y_{0:n} \) the Process \( \{X_k\}_{k \geq 0} \) still is a Markov Chain, although non-homogeneous, with a transition kernel called the Forward Smoothing Kernel. We state the transition kernel here for our case, also conditional on the controls.
**Definition 2** (Definition 3.3.1 in [1]). *Forward Smoothing Kernels.* Given $n \geq 0$ define for indices $k \in \{0, \ldots, n - 1\}$ the transition kernels

$$F_{k|n}(x, A, y_{k+1:n}, u_{k:n}) = \int_A Q^{u_k}(x, dx_{k+1}) g(x_{k+1}, y_{k+1}, \beta_{k+1|n}(x_{k+1})) \beta_{k|n}(x)$$

Note that the Forward Smoothing Kernels are defined in terms of the backward variables.

We are generally interested in calculating smoothers and filters for our POMDP.

**Definition 3** (Definition 3.1.3 in [1]). We let $\phi_{\nu,k|n}$ denote the conditional distribution of $X_{k:l}$ given $Y_{0:n}$ and in our case $u_{0:n-1}$ as well.

The Forward Smoothing Kernel allows us a convenient way of calculating the smoothing distributions. We first compute all the backward variables $\beta_{k|n}$ using the backward recursion given. We then note that $\phi_{\nu,0|n}$ can be calculated as

$$\phi_{\nu,0|n}(A) = \frac{\int_A \nu(dx_0) g(x_0, y_0) \beta_{0|n}(x_0)}{\int \nu(dx_0) g(x_0, y_0) \beta_{0|n}(x_0)}$$

and then we have the following recursion

$$\phi_{\nu,k+1|n}(x) = \int \phi_{\nu,k|n}(dx_k) F_{k|n}(x_k, x) = \phi_{\nu,k|n} F_{k|n}$$

where $F_{k|n}$ are the forward kernels, and the last equation is a short hand way of writing the integral.

Using this recursion repeatedly allows to express the smoother in the following way

$$\phi_{\nu,k|n}[y_{0:n}, u_{0,n-1}] = \phi_{\nu,0|n} \prod_{i=1}^{k} F_{i-1|n}[y_{i:n}, u_{i-1:n-1}]$$

### B.2.3 Total Variation and the Dobrushin Coefficient

To continue towards forgetting properties we need to introduce Total variation (see Definition 4.3.1 in [1]). Let $\xi$ be a signed measure (it can be negative) and let $\xi = \xi_+ - \xi_-$ where $\xi_+, \xi_-$ are (positive) measures. So if $X$ is the state space

$$||\xi||_{TV} = \xi_+(X) + \xi_-(X)$$
Then we need to define the Dobrushin Coefficient (see Definition 4.3.7 in [1]). Let $K$ be a transition Kernel from $X$ to $Y$.

$$\delta(K) = \frac{1}{2} \sup_{(x,x') \in X \times X} ||K(x,\cdot) - K(x',\cdot)||_{TV}$$

The Dobrushin coefficient is sub-multiplicative (see Prop. 4.3.10 in [1]). If $K : X \to Y, R : Y \to Z$ are 2 transition kernels we have

$$\delta(KR) = \delta \left( \int K(\cdot, dx) R(x,\cdot) \right) \leq \delta(K) \delta(R)$$

It can be shown that $0 \leq \delta(K) \leq 1$, however to get forgetting properties we often need $\delta(K) \leq 1 - \varepsilon$, where $\varepsilon > 0$.

The latter inequality holds if we assume the Doeblin Condition is satisfied:

**Assumption 2** (Assumption 4.3.12 in [1]). There exist an integer $m \geq 1$, $\varepsilon \in (0,1)$, and a probability measure $\nu$ on $(X,\mathcal{X})$ such that for any $x \in X$ and $A \in \mathcal{X}$,

$$Q^m(x, A) \geq \varepsilon \nu(A)$$

Under these assumptions Lemma 4.3.13 in [1] gives $\delta(Q^m) \leq 1 - \varepsilon$.

When considering forgetting properties it seems logical to show that the filter $\phi_{\nu,k|n}$ depends less and less on the initial distribution of $X_0 \sim \nu$, as $k$ increases. Specifically we have that when comparing initial distributions $\nu$ and $\nu'$:

$$\phi_{\nu,k|n}(y_{0:n}, u_{0:n-1}, x_k) - \phi_{\nu',k|n}(y_{0:n}, u_{0:n-1}, x_k)$$

$$= \int \cdots \int \left( \phi_{\nu,0|n}(y_{0:n}, u_{0:n-1}, x_k) - \phi_{\nu',0|n}(y_{0:n}, u_{0:n-1}, x_k) \right) \prod_{i=1}^{k} F_{i-1|n}(x_{k-1}, x_k)$$

Now using Corollary 4.3.9 in [1] we have

$$||\xi K - \xi' K||_{TV} \leq \delta(K)||\xi - \xi'||_{TV}$$

where $\xi, \xi'$ are probability measures, $K$ a transition kernel.

Using this on our representation of the filters gives

$$||\phi_{\nu,k|n} - \phi_{\nu',k|n}||_{TV} \leq \delta \left( \prod_{i=1}^{k} F_{i-1|n}(y_{i:n}, \cdot) \right) ||\phi_{\nu,0|n} - \phi_{\nu',0|n}||_{TV}$$
and since the Dobrushin coefficient is sub-multiplicative
\[
\leq \prod_{i=1}^{k} \delta \left(F_{i-1|n}(y_{i:n}, \cdot) \right) \| \phi_{\nu,0|n} - \phi_{\nu',0|n} \|_{TV}
\]
and since the Dobrushin coefficient \( \delta \) satisfies \( 0 \leq \delta \leq 1 \) we at least have that the difference between the 2 filters is non-expanding.

Establishing forgetting properties thus amounts to showing \( \delta(F_{i-1|n}(y_{i:n})) \leq 1 - \varepsilon \) for the forward smoothing kernels \( F_{i|n} \). Note that so far no assumptions have been made on how quickly the Hidden Markov Model mixes. Those assumptions are made to get \( \delta(F_{i|n}) \leq 1 - \varepsilon \).

Cappe et al. \[1\] establish contracting bounds on the Dobrushin coefficient by imposing Strong Mixing conditions on the transition probabilities of the Hidden Markov Model.

**Assumption 3** (Assumption 4.3.21 in \[1\]). *Strong Mixing Conditions*. There exist a transition kernel \( K : Y \to X \) and measurable functions \( \varsigma^- \) and \( \varsigma^+ \) from \( Y \) to \((0, \infty)\) such that for any \( A \in X \) and \( y \in Y \),
\[
\varsigma^-(y) K(y, A) \leq \int_A Q(x, dx') g(x', y) \leq \varsigma^+(y) K(y, A)
\]

In our case we have different transition kernels for each control. The weakest assumptions we can get away with is, if each transition kernel \( Q^u \) has a corresponding transition kernel \( K^u \) and measurable functions \( \varsigma^-(y, u) \) and \( \varsigma^+(y, u) \) satisfying the strong mixing condition. By letting \( \varsigma^-(y) = \min_u \varsigma^-(y, u) \) and \( \varsigma^+(y) = \max_u \varsigma^+(y, u) \) we see that we can consider the same \( \varsigma \) functions for each transition kernel \( Q^u \). We restate the Strong mixing conditions:

**Assumption 4.** *Modified Strong Mixing Conditions*. For each control \( u \) there exist a transition kernel \( K^u : Y \to X \) and measurable functions \( \varsigma^- \) and \( \varsigma^+ \) from \( Y \) to \((0, \infty)\) such that for any \( A \in X \) and \( y \in Y \),
\[
\varsigma^-(y) K^u(y, A) \leq \int_A Q^u(x, dx') g(x', y) \leq \varsigma^+(y) K^u(y, A)
\]

Lemma 4.3.22 in Cappe et al. \[1\] uses the mixing conditions stated above to establish contracting bounds on the Dobrushin coefficient. We restate the lemma for the POMDP case, where we also condition on the controls, and use the modified mixing conditions.

40
Theorem 3 (Lemma 4.3.22 in [1]). Under the strong mixing conditions the following holds

(i) For any non-negative integers \( k \) and \( n \) such that \( k < n \) and \( x \in X \),

\[
\prod_{j=k+1}^{n} \varsigma^{-}(y_j) \leq \beta_{k|n}[y_{k+1:n}, u_{k:n}](x) \leq \prod_{j=k+1}^{n} \varsigma^{+}(y_j)
\]

(ii) For any non-negative integers \( k \) and \( n \) such that \( k < n \) and any probability measures \( \nu \) and \( \nu' \) on \( (X, \mathcal{X}) \),

\[
\frac{\varsigma^{-}(y_{k+1})}{\varsigma^{+}(y_{k+1})} \leq \frac{\int \nu(dx) \beta_{k|n}[y_{k+1:n}, u_{k:n}](x)}{\int \nu'(dx) \beta_{k|n}[y_{k+1:n}, u_{k:n}](x)} \leq \frac{\varsigma^{+}(y_{k+1})}{\varsigma^{-}(y_{k+1})}
\]

(iii) For any non-negative integers \( k \) and \( n \) such that \( k < n \), there exists a transition kernel \( \lambda_{k|n} \) from \( (Y^{n-k}, \mathcal{Y}^{n-k}) \) to \( (X, \mathcal{X}) \) such that for any \( x \in X \), \( A \in \mathcal{X} \), and \( y_{k+1:n} \in Y^{n-k} \),

\[
\frac{\varsigma^{-}(y_{k+1})}{\varsigma^{+}(y_{k+1})} \lambda_{k,n}(y_{k+1:n}, u_{k:n}, A) \leq \frac{\varsigma^{+}(y_{k+1})}{\varsigma^{-}(y_{k+1})} \lambda_{k,n}(y_{k+1:n}, u_{k:n}, A)
\]

(iv) For any non-negative integers \( k \) and \( n \), the Dobrushin coefficient of the forward smoothing kernel \( F_{k|n}[y_{k+1:n}, u_{k:n}] \) satisfies

\[
\delta(F_{k|n}[y_{k+1:n}, u_{k:n}]) \leq \rho_{0}(y_{k+1}) := 1 - \frac{\varsigma^{-}(y_{k+1})}{\varsigma^{+}(y_{k+1})}
\]

if \( k < n \), and

\[
\delta(F_{k|n}[y_{k+1:n}, u_{k:n}]) \leq 1 - \int \varsigma^{-}(y)dy
\]

if \( k \geq n \).

Proof. The proof is the same as for the corresponding lemma in Cappe et al. [1], but with slight modifications to allow for conditioning on controls.
(i) Letting $A = X$ in the strong mixing conditions we find that for all $u$

$$\zeta^-(y) \leq \int Q^n(x, dx') g(x', y) \leq \zeta^+(y)$$

We also have

$$\beta_{k|n}(x) = \int_{x_{k+1}}^{x} \cdots \int_{x_n} Q^{u_k}(x, dx_{k+1}) g(x_{k+1}, y_{k+1}) \prod_{l=k+2}^{n} Q^{u_l-1}(x_{l-1}, dx_l) g(x_l, y_l)$$

$$= \int_{x_{k+1}}^{x} Q^{u_k}(x, dx_{k+1}) g(x_{k+1}, y_{k+1})$$

$$\times \int_{x_{k+2}}^{x} \cdots \int_{x_n} Q^{u_{k+1}}(x_{k+1}, dx_{k+2}) g(x_{k+2}, y_{k+2}) \prod_{l=k+3}^{n} Q^{u_{l-1}}(x_{l-1}, dx_l) g(x_l, y_l)$$

$$\leq \zeta^+(y_{k+1}) \sup_{x_{k+1}} \cdots \sup_{x_n} \beta_{k|n}(x) \leq \prod_{j=k+1}^{n} \zeta^+(y_j)$$

The other inequality is similar.

(ii) Using the recursion for the backward variables we find

$$\int \nu(dx) \beta_{k|n}(y_{k+1:n}, u_{k:n})$$

$$= \int \int_{x_{k+1}}^{x} \nu(dx) Q^{u_k}(x, x_{k+1}) g(x_{k+1}, y_{k+1}) \beta_{k+1|n}(y_{k+2:n}, u_{k+1:n}, dx_{k+1})$$

$$= \int_{x_{k+1}}^{x} \left[ \int_{x} \nu(dx) Q^{u_k}(x, x_{k+1}) g(x_{k+1}, y_{k+1}) \right] \beta_{k+1|n}(y_{k+2:n}, u_{k+1:n}, dx_{k+1})$$

$$\leq \int_{x_{k+1}}^{x} \left[ \int_{x} \nu(dx) \zeta^+(y_{k+1}) K^{u_k}(y_{k+1}, x_{k+1}) \right] \beta_{k+1|n}(y_{k+2:n}, u_{k+1:n}, dx_{k+1})$$

$$= \zeta^+(y_{k+1}) \int_{x_{k+1}}^{x} K^{u_k}(y_{k+1}, x_{k+1}) \beta_{k+1|n}(y_{k+2:n}, u_{k+1:n}, dx_{k+1})$$

We get a similar inequality for $\zeta^-$. Also note that the last integral doesn’t depend on $\nu$, so it cancels when we take the ratio. The result follows.
(iii) We have that
\[ F_{k|n|}[y_{k+1:n}, u_{k:n}](x, A) = \int_A Q^{uk}(x, dx_{k+1}) g(x_{k+1}, y_{k+1}) \beta_{k+1|n}(x_{k+1}) \]
\[ \leq \frac{s^+(y_{k+1})}{s^-(y_{k+1})} \int_A K^{uk}(y_{k+1}, dx_{k+1}) \beta_{k+1|n}(x_{k+1}) \]
and we can set
\[ \lambda_{k|n}(y_{k+1:n}, u_{k:n}, A) = \frac{\int_A K^{uk}(y_{k+1}, dx_{k+1}) \beta_{k+1|n}(x_{k+1})}{\int K^{uk}(y_{k+1}, dx_{k+1}) \beta_{k+1|n}(x_{k+1})} \]

(iv) Using (iii) we find that
\[ F_{k|n|}[y_{k+1:n}, u_{k:n}](x, A) \geq \frac{s^-(y_{k+1})}{s^+(y_{k+1})} \lambda_{k|n}(y_{k+1:n}, u_{k:n}, A) \]
and thus Assumption 4.3.12 holds and Lemma 4.3.13 gives
\[ \delta(F_{k|n}) \leq \rho_0(y_{k+1}) = 1 - \frac{s^-(y_{k+1})}{s^+(y_{k+1})} \]

\[ \square \]

**Theorem 4** (Proposition 4.3.23 in [1]). Under the strong mixing conditions the following holds

(i) We let \( \nu \) and \( \nu' \) be two different initial distributions for \( X_0 \). Now for \( k \leq n \)
\[ ||\phi_{\nu,k|n}[y_{0:n}, u_{0:n-1}] - \phi_{\nu',k|n}[y_{0:n}, u_{0:n-1}]||_{TV} \]
\[ \leq \left[ \prod_{j=1}^{k} \rho_0(y_j) \right] ||\phi_{\nu,0|n}[y_{0:n}, u_{0:n-1}] - \phi_{\nu',0|n}[y_{0:n}, u_{0:n-1}]||_{TV} \]
\[ \leq 2 \left[ \prod_{j=1}^{k} \rho_0(y_j) \right] \]

43
(ii) For any non-negative integers $j, k, n$ such that $j \leq k \leq n$

$$||P_\nu(X_k \in \cdot | y_{0:n}, u_{0:n-1}) - P_\nu(X_k \in \cdot | Y_{j:n}, u_{j:n-1})||_{TV} \leq 2 \prod_{i=j}^k \rho_0(y_i)$$

where $\nu$ is the initial distribution of $X_0$.

**Proof.** (i) Earlier we had

$$||\phi_{\nu, k|n} - \phi_{\nu', k|n}||_{TV} \leq \prod_{i=1}^k \delta(F_{i-1|n}(y_{i:n}, \cdot)) \ ||\phi_{\nu, 0|n} - \phi_{\nu', 0|n}||_{TV}$$

and the first inequality now follows from the Lemma 4.3.22 part (iv). The factor ”2” follows from using the triangle inequality on the difference of two probability measures.

(ii) This is just like part (i) except we consider different initial distributions for $X_j$.

\[\square\]

**B.2.4 Invoking Fisher’s identity, chapter 10 in Cappe et al.**

Fisher’s identity (see Proposition 10.1.6 in [1]) gives an alternative way to calculate the score function $\frac{\partial}{\partial \theta} l(\theta)$. This is based on the theory of the EM algorithm.

In general one can set $f(x; \theta) \equiv f(x, y; \theta)$, the joint pdf of $x, y$. The likelihood for $Y$ is $L(\theta) = \int f(x; \theta)dx$ and $l(\theta) = \log L(\theta)$ the loglikelihood. Set $p(x; \theta) = \frac{f(x, \theta)}{L(\theta)}$, the conditional of $X$ given $Y$.

Now set

$$Q(\theta, \theta') = \int \log f(x; \theta)p(x; \theta')dx = E[\log f(x; \theta)|Y]$$

and

$$H(\theta, \theta') = -\int \log p(x; \theta)p(x; \theta')dx$$

We find that

$$Q(\theta, \theta') = \int \log f(x; \theta)p(x; \theta')dx = \int \log(p(x; \theta)L(\theta))p(x; \theta')dx$$

44
\[
= l(\theta) + \int \log p(x; \theta)p(x; \theta') dx = l(\theta) - H(\theta, \theta')
\]

It is easily seen that \(H(\theta, \theta')\) is minimized as a function of \(\theta\) at \(\theta'\) and thus

\[
\frac{\partial}{\partial \theta} l(\theta') = \frac{\partial}{\partial \theta} Q(\theta, \theta')|_{\theta=\theta'} + \frac{\partial}{\partial \theta} H(\theta, \theta')|_{\theta=\theta'} = \int \frac{\partial}{\partial \theta} \log f(x; \theta)|_{\theta=\theta'} p(x, \theta') dx
\]

assuming we can exchange derivatives with integration. The last equation is called Fisher’s identity.

In the POMDP case this translates to

\[
f(x_0: n, y_0: n, u_0: n - 1, \theta) = \nu(x_0)g(x_0, y_0; \theta)q^{u_0}(x_0, x_1; \theta)g(x_1, y_1; \theta)
\]

\[
\cdots q^{u_{n-1}}(x_{n-1}, x_n; \theta)g(x_n, y_n; \theta)
\]

and then

\[
\log f = \log \nu(x_0; \theta) + \log(x_0, y_0; \theta) + \sum_{k=0}^{n-1} \log(q^{u_k}(x_k, x_{k+1}; \theta)g(x_{k+1}, y_{k+1}; \theta))
\]

and

\[
Q(\theta, \theta') = E[\log f|Y_{0:n}, u_{0:n-1}]
\]

\[
= E_\theta[\log \nu(x_0; \theta)|Y_{0:n}, u_{0:n-1}] + E_\theta[\log g(x_0, y_0; \theta)|Y_{0:n}, u_{0:n-1}]
\]

\[
+ \sum_{k=0}^{n-1} E_\theta[\log(q^{u_k}(x_k, x_{k+1}; \theta)g(x_{k+1}, y_{k+1}; \theta))|Y_{0:n}, u_{0:n-1}]
\]

We set \(\phi(x, x', u, y) = \frac{\partial}{\partial \theta} \log(q^u(x, x'; \theta)g(x', y'; \theta))\) and get

\[
\frac{\partial}{\partial \theta} l(\theta)
\]

\[
= E_\theta[\frac{\partial}{\partial \theta} \log \nu(x_0; \theta)|Y_{0:n}, u_{0:n-1}] + E_\theta[\frac{\partial}{\partial \theta} \log g(x_0, y_0; \theta)|Y_{0:n}, u_{0:n-1}]
\]

\[
+ \sum_{k=0}^{n-1} E_\theta[\phi(x_k, x_{k+1}, u_k, y_{k+1}; \theta)|Y_{0:n}, u_{0:n-1}]
\]

This is a different expression of the score function from what we have used previously.
B.2.5 Bounds on score function, Chapter 12 in Cappe et al.

Set $h_{k,x}(\theta) = \log \left[ \int g(x_k, Y_k) P(X_k \in dx_k | Y_{0:k-1}, u_{0:k-1}, X_0 = x) \right]$. Then our usual loglikelihood is $l_{x,n}(\theta) = \sum_{k=0}^{n} h_{k,x}(\theta)$.

We now wish to use the expression for $\frac{\partial}{\partial \theta} l_{x,n}(\theta)$ derived in the last section. We have that

$$\frac{\partial}{\partial \theta} l_{x,n}(\theta) = \sum_{k=0}^{n} \dot{h}_{k,x}(\theta)$$

but also

$$\frac{\partial}{\partial \theta} l_{x,n}(\theta) = \frac{\partial}{\partial \theta} l_{x,0}(\theta) + \sum_{k=1}^{n} \left\{ \frac{\partial}{\partial \theta} l_{x,k}(\theta) - \frac{\partial}{\partial \theta} l_{x,k-1}(\theta) \right\}$$

This gives an alternative expression of $\dot{h}_{k,x}$. We get $\dot{h}_{0,x}(\theta) = \frac{\partial}{\partial \theta} \log g(x_0, Y_0)$ and for $k \geq 1$

$$\dot{h}_{k,x}(\theta) = \frac{\partial}{\partial \theta} l_{x,k}(\theta) - \frac{\partial}{\partial \theta} l_{x,k-1}(\theta)$$

$$= E \left[ \sum_{i=1}^{k} \phi(X_{i-1}, X_i, Y_i) | Y_{1:k}, u_{0:k-1}, X_0 = x \right]$$

$$- E \left[ \sum_{i=1}^{k-1} \phi(X_{i-1}, X_i, Y_i) | Y_{1:k-1}, u_{0:k-2}, X_0 = x \right]$$

This expression can be generalized to starting the process at other values than zero;

$$\dot{h}_{k,m,x}(\theta) = \log \left[ \int g(x_k, Y_k) P(X_k \in dx_k | Y_{m:k-1}, u_{m:k-1}, X_m = x) \right]$$

$$= E \left[ \sum_{i=m+1}^{k} \phi(X_{i-1}, X_i, Y_i) | Y_{m+1:k}, u_{m:k-1}, X_m = x \right]$$

$$- E \left[ \sum_{i=m+1}^{k-1} \phi(X_{i-1}, X_i, Y_i) | Y_{m+1:k-1}, u_{m:k-2}, X_m = x \right]$$

This is done in Cappe et al. to extend the process to minus infinity ($m \to -\infty$). We don’t extend the process to infinity, but rather think of $m$ as indicating lack of information, that is assuming that the process starts at $X_m$.

We now prove a modified Lemma 12.5.3 where we use the expression developed above.
Theorem 5 (Lemma 12.5.3 in [1] modified). Assuming strong mixing conditions. Then for \( k \geq 1 \) Cappe et al. [1] prove the following inequality in the HMM case:

\[
(E|\dot{h}_{k,-m,x}(\theta) - \dot{h}_{k,\infty}(\theta)|^2)^{1/2} \leq 12 \left( E \left[ \sup_{x,x' \in X} |\phi_\theta(x, x', Y_1)|^2 \right] \right)^{1/2} \frac{\rho(k+m)/2-1}{1-\rho}
\]

We don’t extend the process to \(-\infty\), but rather starting at \( X_0 \) and we prove the following inequality, also for \( k \geq 1 \)

\[
(E|\dot{h}_{k,0,x_0}(\theta) - \dot{h}_{k,m,x}(\theta)|^2)^{1/2} \leq 8 \sup_{x,x' \in X, u \in U, y \in Y} ||\phi_\theta(x, x', y, u)|| \frac{\rho(k-m)/2-1}{1-\rho}
\]

where \( \rho = \max_{y \in Y} \rho_0(y) \) (See Theorem 3).

Proof. From the representation derived above for \( \dot{h} \) we have

\[
\dot{h}_{k,0,x_0}(\theta) = E \left[ \sum_{i=1}^{k} \phi(X_{i-1}, X_i, Y_i, u_{i-1})|Y_{1:k}, u_{0:k-1}, X_0 = x_0 \right]
\]

(1)

\[-E \left[ \sum_{i=1}^{k-1} \phi(X_{i-1}, X_i, Y_i, u_{i-1})|Y_{1:k-1}, u_{0:k-2}, X_0 = x_0 \right]
\]

(2)

and

\[
\dot{h}_{k,m,x}(\theta) = E \left[ \sum_{i=m+1}^{k} \phi(X_{i-1}, X_i, Y_i, u_{i-1})|Y_{m+1:k}, u_{m:k-1}, X_m = x \right]
\]

(3)

\[-E \left[ \sum_{i=m+1}^{k-1} \phi(X_{i-1}, X_i, Y_i, u_{i-1})|Y_{m+1:k-1}, u_{m:k-2}, X_m = x \right]
\]

(4)

Just like in the proof of Lemma 12.5.3 in [1] we match together different pairs of terms within the sums, depending on their index \( i \). More specifically for \( i = k \) we match together the terms where \( i = k \) in (1) and (3). For \( \frac{k+m}{2} \leq i < k \) we match the terms in (1) with (3) and the terms in (2) with those in (4). For \( m + 1 \leq i < \frac{k+m}{2} \) we match terms in (1) with terms in (2) and terms in (3) with those in (4). That leaves \( i \in 1, \ldots, m \) in \( \dot{h}_{k,0,x_0} \) where we match (1) and (2).
If we look at the case where (1) is matched with (3) we have
\[
||E[\phi_\theta(X_{i-1}, X_i, Y_i, u_{i-1})|Y_{m+1:k}, u_{m:k-1}, X_m = x] - E[\phi_\theta(X_{i-1}, X_i, Y_i, u_{i-1})|Y_{1:k}, u_{0:k-1}]||
\]
\[
= |\int_{x_{i-1}} \int_{x_i} \int_{x_{i-1}} \phi_\theta(x_{i-1}, x_i, Y_i, u_i) \cdot F_{i-1}(x_{i-1}, dx_i) \cdot P_\theta(X_{i-1} \in dx_i|Y_{m+1:k}, u_{m:k-1}, X_m = x) \times \delta_x(dx_m) - P_\theta(X_m \in dx_m|Y_{1:k}, u_{0:k-1})||
\]
\[
\leq 2 \sup_{x,x' \in X, u \in U} ||\phi_\theta(x, x', Y_i, u)|| |\rho^{(i-1)-m}
\]
where \(F_{i-1} = F_{i-1}[y_{i:k}, u_{i-1:k}]\) is the Forward Smoothing Kernel, and the inequality stems from Proposition 4.3.23 (i) where the second line can be thought of as two different initial distributions for \(X_m\), and the kernel \(F\) is bounded by 1.

Matching (2) with (4) is similar. For matching (1) with (2) and (3) with (4) we need a "Backwards bound":
\[
||P_\theta(X_i \in \cdot |Y_{m+1:k}, u_{m:k-1}, X_m = x) - P_\theta(X_i \in \cdot |Y_{m+1:k-1}, u_{m:k-2}, X_m = x)||_{TV}
\]
\[
\leq 2\rho^{k-1-i}
\]
that is established below, see Theorem 6. For matching (3) with (4) we get
\[
||E_\theta[\phi_\theta(X_{i-1}, x_i, Y_i, u_{i-1})|Y_{m+1:k}, u_{m:k-1}, X_m = x] - E_\theta[\phi_\theta(X_{i-1}, x_i, Y_i, u_{i-1})|Y_{m+1:k-1}, u_{m:k-2}, X_m = x]||
\]
\[
= |\int_{x_{i-1}} \int_{x_i} \phi_\theta(x_{i-1}, x_i, Y_i, u_{i-1}) \cdot B_i(x_i, dx_{i-1}) \times P_\theta(X_i \in dx_i|Y_{m+1:k}, u_{m:k-1}, X_m = x) - P_\theta(X_i \in dx_i|Y_{m+1:k-1}, u_{m:k-2}, X_m = x)||
\]
\[
\leq 2 \sup_{x,x' \in X, u \in U} ||\phi_\theta(x, x', Y_i, u)|| |\rho^{(k-1)-i}
\]
where \(B_i\) is the Backwards Smoothing Kernel described below. Matching (1) with (2) is a special case of the above.

Going back to our original objective, we have
\[
\left( E_\theta||\hat{h}_{k,m,x}(\theta) - \hat{h}_{k,0,x_0}(\theta)||^2 \right)^{1/2} = \left( E||\sum a_i||^2 \right)^{1/2}
\]
where \(\sum a_i\) is a sum over the pairs we considered above. Now by Minkowski’s inequality we have
\[
\leq \sum \left( E||a_i||^2 \right)^{1/2}
\]
Now we have that \( ||a_i|| \leq 2 \sup_{x,x' \in X, u \in U} ||\phi_{\theta}(x, x', Y_i, u)|| \rho^{b_i} \) where \( b_i \) is the power of \( \rho \) associated with \( a_i \).

\[
\leq \sum 2 \left( \mathbb{E} \sup_{x,x' \in X, u \in U} ||\phi_{\theta}(x, x', Y_i, u)||^2 \right)^{1/2} \rho^{b_i}
\]

At this point Cappe et al. [1] argue that since in their case the process was started at infinity and the process is homogeneous the expected value over \( Y_i \) is always the same by stationarity, and \( Y_i \) can be exchanged by \( Y_1 \). Since arguing for stationarity is more of a stretch for us, we also take the supremum over \( Y \) and remember that that set is also finite.

\[
\leq 2 \left( \sup_{x,x' \in X, u \in U, y \in Y} ||\phi_{\theta}(x, x', y, u)|| \right)^{1/2} \sum \rho^{b_i}
\]

We now deal with the sum of \( \rho \) to different powers.

From \( i = k \) we have \( \rho^{k-1-m} \) where we matched (1) with (3). For \( \frac{k+m}{2} \leq i < k \) we have \( 2\rho^{i-1-m} \) where we matched (1) with (3) and (2) with (4). For \( m+1 \leq i < \frac{k+m}{2} \) we have \( 2\rho^{k-1-i} \) from matching (1) with (2) and (3) with (4). Finally for \( 1 \leq i \leq m \) we have \( \rho^{k-1-i} \) from matching (1) with (2). This gives

\[
\sum \rho^{b_i} = \rho^{k-1-m} + \sum_{i=\frac{(k+m)}{2}}^{k-1} 2\rho^{i-1-m} + \sum_{i=m+1}^{(k+m)/2-1} 2\rho^{k-1-i} + \sum_{i=1}^{m} \rho^{k-1-i}
\]

\[
\leq 2 \sum_{i=\frac{(k+m)}{2}}^{\infty} \rho^{i-1-m} + 2 \sum_{i=-\infty}^{(k+m)/2-1} \rho^{k-1-i}
\]

\[
= 2\frac{\rho^{(k-m)/2-1}}{1 - \rho} + 2\frac{\rho^{(k-m)/2}}{1 - \rho} \leq 4\frac{\rho^{(k-m)/2-1}}{1 - \rho}
\]

Thus, finally we have

\[
\left( \mathbb{E}_\theta ||\hat{h}_{k,m,x}(\theta) - \hat{h}_{k,0,x_0}(\theta)||^2 \right)^{1/2} \leq 8 \sup_{x,x' \in X, u \in U, y \in Y} ||\phi_{\theta}(x, x', y, u)|| \frac{\rho^{(k-m)/2-1}}{1 - \rho}
\]

\[\square\]
Theorem 6 (Proposition 12.5.4 modified).

\[ ||P_\theta(X_i \in \cdot | Y_{m+1:k}, u_{m:k-1}, X_m = x) - P_\theta(X_i \in \cdot | Y_{m+1:k-1}, u_{m:k-2}, X_m = x) ||_{TV} \leq 2 \rho^{k-1-i} \]

Proof. The idea behind this proof is to replicate all the results derived so far for the Backward Smoothing Kernel. That is, conditional on \( Y_{m+1:k}, u_{m:k-1} \) and \( X_m = x_m \) the time-reversed process \( X \) is a non-homogeneous Markov Chain, where the conditional probability of moving from \( X_{j+1} \) to \( X_j \) given all the observations \( Y_{m+1:j-1}, u_{m:j-2} \) and initial condition ends up only depending on \( Y_{m+1:j}, u_{m:j} \) and the initial condition, and is governed by the Backwards Smoothing Kernel given by

\[ B_{x_{m,j}[u_{m+1:j}, u_{m:j}]}(x, f) \]

\[ = \int \cdots \int \prod_{r=m+1}^j Q_{u_{r-1}}(x_r, dx_r)g(x_r, y_r)f(x_j)Q_{u_{j}}(x_j, x) \]

\[ \int \cdots \int \prod_{r=m+1}^j Q_{u_{r-1}}(x_r, dx_r)g(x_r, y_r)Q_{u_{j}}(x_j, x) \]

Just as we did in Lemma 4.3.22 we can show

\[ \frac{\varsigma^-(y_j)}{\varsigma^+(y_j)} \nu_{x_{m,j}[y_{m+1}, u_{m:j}]} \]

\[ \leq B_{x_{m,j}[y_{m+1:j}, u_{m:j}]}(x_j, \cdot) \]

\[ \frac{\varsigma^+(y_j)}{\varsigma^-(y_j)} \nu_{x_{m,j}[y_{m+1}, u_{m:j}]} \]

where

\[ \nu_{x_{m,j}[y_{m+1}, u_{m:j}]}(f) \]

\[ = \int \cdots \int \prod_{r=m+1}^j Q_{u_{r-1}}(x_r, dx_r)g(x_r, y_r)f(x_j) \]

\[ \int \cdots \int \prod_{r=m+1}^j Q_{u_{r-1}}(x_r, dx_r)g(x_r, y_r) \]

As we showed there this gives

\[ \delta(B_{x_{m,j}}) \leq 1 - \frac{\varsigma^-(y_j)}{\varsigma^+(y_j)} \]

We now get that the 2 smoothers we are interested in can be thought of as smoothers of the reversed Markov Chain from \( k - 1 \) to \( m \) with 2 different initial distributions for \( X_{k-1} \), the starting position. We get

\[ ||P_\theta(X_i \in \cdot | Y_{m+1:k}, u_{m:k-1}, X_m = x) - P_\theta(X_i \in \cdot | Y_{m+1:k-1}, u_{m:k-2}, X_m = x) ||_{TV} \]

50
\[
\leq \|P_\theta(X_{k-1} \in \cdot | Y_{m+1:k}, u_{m:k-1}, X_m = x) - P_\theta(X_{k-1} \in \cdot | Y_{m+1:k-1}, u_{m:k-2}, X_m = x)\|_{TV} \\
\times \prod_{j=i+1}^{k-1} \delta(B_{x_m,j}) \\
\leq 2 \prod_{j=i+1}^{k-1} \rho_0(y_j) \leq 2\rho^{k-1-i}
\] 

(Where \(\rho = \max_{y \in Y} \rho_0(y)\)) \qed