A Separation-based Approach to Data-based Control for Large-Scale Partially Observed Systems

Dan Yu¹, Mohammadhussein Rafieisakhaei² and Suman Chakravorty¹

Abstract—This paper studies the partially observed stochastic optimal control problem for systems with state dynamics governed by partial differential equations (PDEs) that leads to an extremely large problem. First, an open-loop deterministic trajectory optimization problem is solved using a black-box simulation model of the dynamical system. Next, a Linear Quadratic Gaussian (LQG) controller is designed for the nominal trajectory-dependent linearized system which is identified using input-output experimental data consisting of the impulse responses of the optimized nominal system. A computational nonlinear heat example is used to illustrate the performance of the proposed approach.

I. INTRODUCTION

Stochastic optimal control problems, also known as Markov decision problems (MDPs), have found numerous applications in the Sciences and Engineering. In general, the goal is to control a stochastic system subject to transition uncertainty in the state dynamics so as to minimize the expected running cost of the system. The MDPs are termed Partially Observed (POMDP) if there is sensing uncertainty in the state of the system in addition to the transition uncertainty. In this paper, we consider the stochastic control of partially observed nonlinear dynamical systems that are governed by partial differential equations (PDE). In particular, we propose a novel data based approach to the solution of very large POMDPs wherein the underlying state space is obtained from the discretization of a PDE: problems whose solution has never been hitherto attempted using approximate MDP solution techniques.

It is well known that the global optimal solution for MDPs can be found by solving the Hamilton-Jacobi-Bellman (HJB) equation [1]. The solution techniques can be further divided into model based and model free techniques, according as whether the solution methodology uses an analytical model of the system or it uses a black box simulation model, or actual experiments.

In model based techniques, many methods [2] rely on a discretization of the underlying state and action space, and hence, run into the "curse of dimensionality (COD)", the fact that the computational complexity grows exponentially with the dimension of the state space of the problem. The most computationally efficient among these techniques are trajectory-based methods, first described in [3]. These methods expand the nonlinear system equations about a deterministic nominal trajectory, and perform a localized version of policy iteration to iteratively optimize the trajectory. For example, the differential dynamic programming (DDP) [4, 5] linearizes the dynamics and the cost-to-go function around a given nominal trajectory, and designs a local feedback controller using DP. The iterative Linear Quadratic Gaussian (ILQG) [6, 7], which is closely related to DDP, considers the first order expansion of the dynamics (in DDP, a second order expansion is considered), and designs the feedback controller using Riccati-like equations, and is shown to be computationally more efficient. In both approaches, the control policy is executed to compute a new nominal trajectory, and the procedure is repeated until convergence.

In the model free solution of MDPs, the most popular approaches are the adaptive dynamic programming (ADP) [8, 9] and reinforcement learning (RL) paradigms [10, 11]. They are essentially the same in spirit, and seek to improve the control policy for a given black box system by repeated interactions with the environment, while observing the system’s responses. The repeated interactions, or learning trials, allow these algorithms to construct a solution to the DP equation, in terms of the cost-to-go function, in an online and recursive fashion. Another variant of RL techniques is the so-called Q-learning method, and the basic idea in Q-learning is to estimate a real-valued function \( Q(x,a) \) of states and actions instead of the cost-to-go function \( V(x) \). For continuous state and control space problems, the cost-to-go functions and the Q-functions are usually represented in a functionally parameterized form, for instance, in the linearly parametrized form \( Q(x,a) = \theta^\top \phi(x,a) \), where \( \theta \) is the unknown parameter vector, and \( \phi \) is a pre-defined basis function, \( (\cdot)^\top \) denotes the transpose of \( (\cdot) \). Multi-layer neural networks may also be used as nonlinearly parameterized approximators instead of the linear architecture above. The ultimate goal of these techniques is the estimation/learning of the parameters \( \theta \) from learning trials/ repeated simulations of the underlying system. However, the size of the parameter \( \theta \) grows exponentially in the size of the state space of the problem without a compact parametrization of the cost-to-go or Q function in terms of the a priori chosen basis functions for the approximation, and hence, these techniques are typically subject to the curse of...
dimensionality. Albeit a compact parametrization may exist, a priori, it is usually never known.

In the past several years, techniques based on the differential dynamic programming/ LQG approach [4, 5, 12, 13], such as the RL techniques [14, 15] have shown the potential for RL algorithms to scale to higher dimensional continuous state and control space problems, in particular, high dimensional robotic task planning and learning problems. These methods are a localized version of the policy search [16, 17, 18, 19] technique that seek to directly optimize the feedback policy via a compact parameterization. For continuous state and control space problems, the method of choice is to wrap an LQR feedback policy around a nominal trajectory and then perform a recursive optimization of the feedback law, along with the underlying trajectory, via repeated simulations/iterations. However, the parametrization can still be very large for partially observed problems (at least \(O(d^2)\)) where \(d\) is the dimension of the state space) or large motion planning problems such as systems governed by partial differential equations wherein the (discretized) state is very high dimensional (thousands/ millions of states) which are typically partially observed thereby compounding the problem. Furthermore, there may be convergence problems with these techniques that can lead to the so-called “policy chatter” phenomenon [14].

Fundamentally, rather than solve the derived “Dynamic Programming” problem as in the majority of the approaches above that requires the optimization of the feedback law, our approach is to directly solve the original stochastic optimization problem in a “separated open loop -closed loop” fashion wherein: 1) we solve an open loop deterministic optimization problem to obtain an optimal nominal trajectory in a model free fashion, and then 2) we design a closed loop controller for the resulting linearized time-varying system around the optimal nominal trajectory, again in a model free fashion. Nonetheless, the above “divide and conquer” strategy can be shown to be near optimal.

The primary contributions of the proposed approach are as follows:

1) We specify a detailed set of experiments to accomplish the closed loop controller design for any unknown nonlinear system, no matter how high dimensional. This series of experiments consists of a sequence of input perturbations to collect the impulse responses of the system, first to find an optimized nominal trajectory, and then to recover the LTV system corresponding to the perturbations of the nominal system in order to design an LQG controller for the LTV system.

2) In general, for large scale systems with partially observed states, the system identification algorithm such as time-varying ERA [20] automatically constructs reduced order model (ROM) of the LTV system, and hence, results in a reduced order estimator and controller. Therefore, even for large scale systems such as partially observed systems with the state dynamics governed by PDEs, the computation of the feedback controller is nevertheless computationally tractable, for instance, in the partially observed nonlinear heat control problem considered in this paper, the complexity is reduced by \(O(10^9)\) when compared to DDP based RL techniques.

3) We provide a unification of traditional linear and nonlinear optimal control techniques with ADP and RL techniques in the context of Stochastic Dynamic Programming problems.

The rest of the paper is organized as follows. In Section II the basic problem formulation is outlined. In Section III we propose a separation based stochastic optimal control algorithm, with discussions of implementation problems. In Section IV we test the proposed approach using a one-dimensional nonlinear heat problem.

II. PROBLEM SETUP

Consider a discrete time nonlinear dynamical system:

\[ x_{k+1} = f(x_k, u_k, w_k), \]
\[ y_k = h(x_k, v_k), \]

where \(x_k \in \mathbb{R}^{n_x}, y_k \in \mathbb{R}^{n_y}, u_k \in \mathbb{R}^{n_u}\) are the state vector, the measurement vector and the control vector at time \(k\) respectively. The system function \(f(\cdot)\) and measurement function \(h(\cdot)\) are nonlinear. The process noise \(w_k\) and measurement noise \(v_k\) are assumed as zero-mean, uncorrelated Gaussian white noise, with covariance \(W\) and \(V\) respectively. In considering PDEs, the dynamics above are the discretized version of the equations (using Finite Difference (FD) or Finite Element (FE) schemes). Typically, the discretization leads to a very large state space problem consisting of at least hundreds of states and typically millions of states for larger problems.

The belief \(b(x_k)\) is defined as the distribution of the state \(x_k\) given all past control inputs and sensor measurements, and is denoted by \(b_k\). In this paper, we represent beliefs by Gaussian distributions, and denote the belief \(b_k = \mu_k, \Sigma_k\), where \(\mu_k\) and \(\Sigma_k\) are the mean and covariance of the Gaussian belief state. Denote the belief dynamics

\[ b_{k+1} = \tau(b_k, u_k, y_{k+1}), \]

and assume that \(b_0\) is known. Note that if the belief is Gaussian, the belief state is \(O(n_x^2)\), which for a PDE is extremely large due to the fact that \(n_x\) is very large.

In this paper, we consider the following stochastic optimal control problem.

Stochastic Control Problem: For the system with unknown nonlinear dynamics, i.e., \(f(\cdot)\) and \(h(\cdot)\) are unknown, the optimal control problem is to find the control policies \(\pi = \{\pi_0, \pi_1, \ldots, \pi_{N-1}\}\) in a finite time horizon \([0, N]\), where \(\pi_k\) is the control policy at time \(k\), i.e., \(u_k = \pi_k(b_k)\), such that for a given initial belief state \(b_0\), the cost function

\[ J_x = E\left( \sum_{k=0}^{N-1} c_k(b_k, u_k) + c_N(b_N) \right), \]

is minimized, where \(\{c_k(\cdot, \cdot)\}_{k=0}^{N-1}\) denotes the immediate cost function, and \(c_N(\cdot)\) denotes the terminal cost. The expectation is taken over all randomness.

III. SEPARATION BASED FEEDBACK CONTROL DESIGN

The stochastic control problem is solved in a separated open loop- closed loop (SOC) fashion, i.e., first, we solve a noiseless open-loop optimization problem to find a nominal
optimal trajectory and then design a linearized closed-loop controller around the nominal trajectory, such that, with existence of stochastic perturbations, the state stays close to the optimal open-loop trajectory. The three-step framework to solve the stochastic feedback control problem may be summarized as follows.

- Solve the open loop optimization problem using a general nonlinear programming (NLP) solver with a black box simulation model of the dynamics, where the belief dynamics is updated using an Ensemble Kalman Filter (EnKF) [21].
- Linearize the system around the nominal open loop optimal belief trajectory, and identify the linearized time-varying system from input-output experiment data using a suitable system identification algorithm such as the time-varying eigensystem realization algorithm (ERA) [20].
- Design an LQG controller which results in an optimal linear control policy around the nominal trajectory.

In the following section, first, we present the “Separation” theorem and then discuss each of the above steps.

A. A Separation Result

Nominal trajectories: Denote \( \{\tilde{u}_k\}_{k=0}^{N-1}, \{\tilde{\mu}_k\}_{k=0}^N \) as the nominal control and state trajectories of the system, respectively. \( \{\bar{y}_k\}_{k=0}^N \) as the corresponding observations and \( \{b_k\}_{k=0}^N \) as the belief trajectories, where:
\[
\tilde{\mu}_{k+1} = f(\tilde{\mu}_k, \tilde{u}_k, 0), \quad \bar{y}_k = h(\tilde{\mu}_k, 0), \\
\tilde{u}_k = \pi_k(\tilde{b}_k), \quad \tilde{b}_{k+1} = \tau(\tilde{b}_k, \tilde{u}_k, \bar{y}_{k+1}),
\]
with the initial conditions of \( \tilde{b}_0 = b_0 \), and \( \bar{\mu}_0 = \mathbb{E}[b_0] \).

Nominal cost function: The nominal cost and its first order expansion are given by (please see [22] for details):
\[
\bar{J} := \sum_{k=0}^{N-1} c_k(\bar{b}_k, \bar{u}_k) + c_N(\bar{b}_N), \\
J \approx \bar{J} + \sum_{k=0}^{N-1} (C_k^b(b_k - \bar{b}_k) + C_k^u(u_k - \bar{u}_k)) + C_N^b(b_N - \bar{b}_N)
\]
(6)

Theorem 1 (Cost Function Linearization Error): The expected first-order linearization error of the cost function is zero, \( \mathbb{E}[\delta J] = 0 \).

B. Open Loop Trajectory Optimization in Belief Space

Consider the following open loop belief state optimization problem with given initial belief state \( b_0 \):
\[
\{u^*_k\}_{k=0}^{N-1} = \arg \min_{\{u_k\}_{k=0}^N} J(\{b_k\}_{k=0}^N, \{u_k\}_{k=0}^{N-1}),
\]
\[
b_{k+1} = \tau(b_k, u_k, \bar{y}_{k+1}),
\]
(7)
where the nominal observations \( \bar{y}_k \) are generated as follows:
\[
x_{k+1} = f(x_k, u_k, 0), \quad \bar{y}_k = h(x_k, 0),
\]
(8)
with \( x_0 = \mu_0 \). Note that given the nominal observations \( \bar{y}_k \), the belief evolution is deterministic and hence, the above is a deterministic optimization problem (this was first posed in the reference [23] in the context of an Extended Kalman Filter (EKF) based belief propagation scheme).

1) Belief Propagation using EnKF: Given an initial belief state \( b_0 = (\mu_0, \Sigma_0) \), and a control sequence \( \{u_k\} \), the EnKF algorithm can be used to propagate the belief state using only a simulator of the state dynamics. This is necessary since we typically do not have access to even an (approximate) belief state simulator for large scale systems such as PDEs, and hence, need to construct one from a state space simulator. Note that the state space dimension is at least in the hundreds for such systems, and thus, a particle filter would suffer from particle depletion, and hence, cannot be used.

Denote the EnKF algorithm as \( \{b_k\}_{k=0}^N = \text{EnKF}(b_0, \{u_k\}_{k=0}^{N-1}) \).
(9)
The details of EnKF can be found in [24], and is briefly summarized in Appendix A; in short, it is a particle filter that is free from the particle depletion problem and is typically used in the filtering of large scale systems such as those governed by PDEs, for instance, meteorological phenomenon.

2) Open loop optimization approach: The open loop optimization problem is solved using the gradient descent approach [25, 26] utilizing an EnKF. Denote the initial guess of the control sequence as \( U^{(0)} = \{u_k^{(0)}\}_{k=0}^{N-1} \), and the corresponding belief state \( B^{(0)} = \{b_k^{(0)}\}_{k=0}^N = \text{EnKF}(b_0, U^{(0)}) \).

The control policy is updated iteratively via
\[
U^{(n+1)} = U^{(n)} - \alpha \nabla_U \bar{J}(B^{(n)}, U^{(n)}),
\]
(10)
until a convergence criterion is met, where \( U^{(n)} = \{u_k^{(n)}\}_{k=0}^{N-1} \) denotes the control sequence in the \( n \)-th iteration, \( B^{(n)} = \{b_k^{(n)}\}_{k=0}^N \) denotes the corresponding belief state, and \( \alpha \) is the step size parameter.

The gradient vector is defined as:
\[
\nabla_U \bar{J}(B^{(n)}, U^{(n)}) = \left( \frac{\partial \bar{J}}{\partial u_N}, \ldots, \frac{\partial \bar{J}}{\partial u_1}, \frac{\partial \bar{J}}{\partial u_0} \right) |_{B^{(n)}, U^{(n)}},
\]
(11)
and without knowing the explicit form of the cost function, each partial derivative with respect to the \( i \)-th control variable
Consider system (13) with zero noise and $\delta x_0 = 0$, the input-output relationship is given by:

$$
\delta y_k = \sum_{j=0}^{k-1} h_{k,j} \delta u_j,
$$

(15)

where $h_{k,j}$ is defined as the generalized Markov parameters, and

$$
\left\{
\begin{array}{ll}
C_k A_{k-1} A_{k-2} \cdots A_{j+1} B_j, & \text{if } j < k - 1, \\
C_k B_{k-1}, & \text{if } j = k - 1, \\
0, & \text{if } j > k - 1.
\end{array}
\right.
$$

(16)

1) Partial Realization Problem \cite{27, 28}: Given a finite sequence of Markov parameters $h_{k,j} \in \mathbb{R}^{n_y \times n_r}$, $k = 1, 2, \cdots, s$, $j = 0, 1, \cdots, k$, the partial realization problem consists of finding a positive integer $n_r$ and LTV system $(A_k, \tilde{B}_k, \tilde{C}_k)$, where $A_k \in \mathbb{R}^{n_r \times n_r}$, $\tilde{B}_k \in \mathbb{R}^{n_r \times n_u}$, $\tilde{C}_k \in \mathbb{R}^{n_y \times n_r}$, such that the identified generalized Markov parameters $\tilde{h}_{k,j} = C_k A_{k-1} A_{k-2} \cdots A_{j+1} B_j = h_{k,j}$. Then $(A_k, \tilde{B}_k, \tilde{C}_k)$ is called a partial realization of the sequence $h_{k,j}$.

We solve the partial realization problem using the time-varying ERA. Time-varying ERA starts by estimating the generalized Markov parameters using input-output experiments, constructs a generalized Hankel matrix, and solves the singular value decomposition (SVD) problem of the constructed Hankel matrix. The details of the time-varying ERA can be found in \cite{20}, and is briefly summarized here.

Define the generalized Hankel matrix as:

$$
\begin{pmatrix}
\tilde{H}_{k}^{(p,q)} & \vdots \\
\vdots & \tilde{H}_{k}^{(p,q)}
\end{pmatrix}
$$

where $p$ and $q$ are design parameters could be tuned for best performance. Denote the rank of the Hankel matrix $\tilde{H}_{k}^{(p,q)}$ is $n_r$, then $p_{n_y} \geq n_r$, $q_{n_u} \geq n_r$.

Given the generalized Markov parameters, we construct two Hankel matrices $\tilde{H}_{k}^{(p,q)}$ and $\tilde{H}_{k+1}^{(p,q)}$, and then solve the singular value decomposition problem:

$$
\tilde{H}_{k}^{(p,q)} = \sum_{r=1}^{r_{k+1}/2} s_{k}^{1/2} \tilde{R}_{k}^{(r,q)}
$$

(18)

where the rank of the Hankel matrix $\tilde{H}_{k}^{(p,q)}$ is $n_r$ and $n_r \leq n_x$.

Then $\Sigma_k \in \mathbb{R}^{n_r \times n_r}$ is the collection of all non-zero singular values, and $U_k \in \mathbb{R}^{n_u \times n_r}$, $V_k \in \mathbb{R}^{n_y \times n_r}$ are the corresponding left and right singular vectors.

Similarly, $\tilde{H}_{k+1}^{(p,q)} = O_{k+1}^{(p)} R_{k+1}^{(q)}$

Then the identified system using time-varying ERA is:

$$
\begin{pmatrix}
\tilde{A}_{k+1} & \tilde{B}_{k+1} & \tilde{C}_{k+1}
\end{pmatrix}
$$

$$
\begin{pmatrix}
O_{k+1}^{(p)} & \tilde{R}_{k+1}^{(q)}
\end{pmatrix}
$$

(19)

\begin{algorithm}
\caption{Open Loop Optimization Algorithm}
\begin{algorithmic}[1]
\Require Start belief $b_0$, cost function $J(\cdot)$, initial guess $U^{(0)} = \{u^{(0)}_{k}\}_{k=0}^{N-1}$, gradient descent design parameters $\alpha, h, \epsilon$.
\Ensure Optimal control sequence $\{\bar{u}_k\}_{k=0}^{N-1}$, belief nominal trajectory $\{\bar{b}_k\}_{k=0}^{N-1}$.
\State $n = 0$, set $\nabla_{U} J(b^{(0)}, U^{(0)}) = \epsilon$.
\While{$\nabla_{U} J(b^{(n)}, U^{(n)}) \geq \epsilon$}
\State Compute the belief $b^{(n)} = EnKF(b^{(0)}, U^{(n)})$.
\State Evaluate the cost function $J(b^{(n)}, U^{(n)})$.
\State Perturb each control variable $u^{(n)}_{i}$ by $h$ and compute the belief $b^{(n)}_{i}$, $i = 0, \ldots, N - 1$, calculate the gradient vector $\nabla_{U} J(b^{(n)}, U^{(n)})$.
\State Update the control policy $U^{(n+1)} = U^{(n)} - \alpha \nabla_{U} J(b^{(n)}, U^{(n)})$.
\State $n = n + 1$.
\EndWhile
\State $\{\bar{u}_k\}_{k=0}^{N-1} = U^{(n)}$, $\{\bar{b}_k\}_{k=0}^{N-1} = EnKF(b^{(0)}, U^{(n)})$.
\end{algorithmic}
\end{algorithm}
where \((.)^+\) denotes the pseudo inverse of \((.)\), \(O_{k+1}^{(p)}\) contains the first \((p-1)n_y\) rows of \(O_k^{(p)}\), and \(O_k^{(p)}\) contains the last \((p-1)n_y\) rows of \(O_k^{(p)}\). Here, we assume that \(n_r\) is constant through the time period of interest, which could also be relaxed.

2) Identify the Generalized Markov Parameters using Input-Output Experiments: Now the problem is how to estimate the generalized Markov parameters. Consider the input-output map for system (13) with zero noise and \(\delta x_0 = 0\):

\[
\delta y_k = \sum_{j=0}^{k-1} h_{k,j} \delta u_j. 
\]

We run \(M\) simulations and in the \(i^{th}\) simulation, choose input sequence \(\{\delta u_{i,(i)}\}_{i=0}^{k}\), and collect the output \(\delta y_{k,(i)}\). The subscript \((i)\) denotes the experiment number. Then the generalized Markov parameters \(\{h_{k,j}\}_{j=0}^{k}\) could be recovered via solving the least squares problem:

\[
\begin{bmatrix}
\delta y_{k,(1)} & \delta y_{k,(2)} & \cdots & \delta y_{k,(M)} \\
0 & h_{k,k-1} & h_{k,k-2} & \cdots & h_{k,0}
\end{bmatrix} \times 
\begin{bmatrix}
\delta u_{k,(1)} & \delta u_{k,(2)} & \cdots & \delta u_{k,(M)} \\
\delta u_{k-1,(1)} & \delta u_{k-1,(2)} & \cdots & \delta u_{k-1,(M)} \\
\vdots & \vdots & \ddots & \vdots \\
\delta u_{0,(1)} & \delta u_{0,(2)} & \cdots & \delta u_{0,(M)}
\end{bmatrix} = 1
\]

where \(M\) is a design parameter and is chosen such that the least squares solution is possible.

Notice that we cannot perturb the system (13) directly. Instead, we identify the generalized Markov parameters as follows.

Run \(M\) parallel simulations with the noise-free system:

\[
x_{k+1,(i)} = f(x_{k,(i)}, \hat{u}_k + \delta u_{k,(i)}, 0),
\]

\[
y_{k,(i)} = h(x_{k,(i)}, 0),
\]

where \(i = 1,2,\cdots, M\), and therefore,

\[
\delta y_{k,(i)} = y_{k,(i)} - h(\hat{x}_{k,0}, 0).
\]

The time-varying ERA used in this paper is summarized in Algorithm 2. Denote the identified deviation system

\[
\delta a_{k+1} = \hat{A}_k \delta a_k + \hat{B}_k (\delta u_k + w_k),
\]

\[
\delta y_k = \hat{C}_k \delta a_k + v_k,
\]

where \(\delta a_k \in \mathbb{R}^{n_r}\) denotes the reduced order model (ROM) deviation states. The dimension \(n_r\) of the ROM is such that \(n_r << n_x\), where \(n_x\) is the dimension of the state, thereby automatically providing a compact parametrization of the problem (please also see Section IV).

D. Closed Loop Controller Design

Given the identified deviation system (24), we design the closed-loop controller to follow the optimal nominal trajectory, which is to minimize the cost function

\[
J_f = \sum_{k=0}^{N-1} (\delta \hat{a}_k^T Q_k \delta a_k + \delta u_k^T R_k \delta u_k) + \delta \hat{a}_N^T Q_N \delta a_N,
\]

where \(\delta \hat{a}_k\) denotes the estimates of the deviation state \(\delta a_k\), \(Q_k, Q_N\) are positive definite, and \(R_k\) is positive semi-definite.

Algorithm 2 LTV System Identification

**Require:** Nominal Trajectory \(\{u_k\}_{k=0}^{N-1}, \{\hat{b}_k\}_{k=0}^{N}\), design parameters \(M, p, q\)

**Ensure:** \(\{\hat{A}_k, \hat{B}_k, \hat{C}_k\}\)

1: \(k = 0\)

2: \(\text{while } k \leq N - 1 \text{ do}\)

3: \(\text{Identify generalized Markov parameters with input and output experimental data using (21), (22) and (23).}\)

4: \(\text{Construct the generalized Hankel matrices } H_{k+1}^{(p,q)}, H_{k+1}^{(p,q)} \text{ using (17).}\)

5: \(\text{Solve the SVD problem, and construct } \{\hat{A}_k, \hat{B}_k, \hat{C}_k\}\) using (19).

6: \(k = k + 1\).

7: \(\text{end while}\)

Algorithm 3 Separation based Stochastic Feedback Control

1: \(\text{Solve the deterministic open-loop optimization problem using Algorithm 1.}\)

2: \(\text{Identify the LTV system using Algorithm 2.}\)

3: \(\text{Solve the decoupled Ricci equations (37), (38) using LTV system for feedback gain } \{L_k\}_{k=0}^{N}.\)

4: \(\text{Set } k = 0, \text{ given initial estimates } \delta \hat{a}_0 = 0, P_0.\)

5: \(\text{while } k \leq N - 1 \text{ do}\)

6: \(u_k = \hat{u}_k - L_k \delta \hat{a}_k, x_{k+1} = f(x_k, u_k, w_k), y_{k+1} = h(x_{k+1}, v_{k+1}).\)

7: \(k = k + 1.\)

8: \(\text{end while}\)

For the linear system (24), the “separation principle” of linear control theory (not the Separation result of Section III-A) can be used (11). Using this result, the design of the optimal linear stochastic controller can be separated into the decoupled design of an optimal Kalman filter and a fully observed optimal LQR controller. The details of the design is standard (11) and is shown briefly in Appendix B.

A flow chart for the Separation based Nonlinear Stochastic Control Design is shown in Fig. (1) and the algorithm is present in Algorithm 3.

E. Discussion

**Direct Data based Controller design:** As mentioned in data-based LQG (29), (30) and data-driven MPC control (31), (32), the linear system \((\hat{A}_k, \hat{B}_k, \hat{C}_k)\) need not be identified to design the LQG controller which can be directly designed from the identified Markov parameters.

**Replanning:** The proposed approach is theoretically valid under a small noise assumption (it is typically valid for medium noise). In practice, due to non-linearities and unknown perturbations, the actual state might deviate from the nominal trajectory during execution whence a replanning starts from the current state in a model predictive control (MPC) fashion. However, unlike in MPC, the replanning does not need to be
done at every time step, only when necessary which is, in general, very infrequently.

Optimality: The open loop law generated by the gradient descent can be guaranteed to be locally optimal under usual regularity conditions. Theorem 1 shows that, under a linear approximation, the stochastic cost is the same as the nominal cost and therefore, locally optimal as well. ILQG/DPD based methods can also only make a claim regarding the local optimality of the nominal control law unlike global policy iteration, and therefore, the guarantees regarding optimality are the same for both.

Complexity: The model free open loop optimization problem has complexity $O(n_u)$, where $n_u$ is the number of inputs, the LTV system identification step is again $O(n_u n_y)$, and the LQG feedback design has complexity $O(n_x^2)$, where $n_x$ is the order of the ROM from the LTV system identification step. Suppose we were to use an ILQG based design such as in [14, 15], the complexity of the controller/policy parametrization is $O(n_u n_x^2)$. Moreover, the policy evaluation step would require an estimation of a parameter of the size $O(n_x^2)$. Since $n_x << n_u$, typically, the complexity of our separated technique is several orders of magnitude smaller (please see Section IV also).

IV. EXPERIMENTS

We test the method on a one-dimensional nonlinear heat transfer problem. The heat transfer along a slab is governed by the partial differential equation:

$$\frac{\partial T}{\partial t} = K(x, T) \frac{\partial^2 T}{\partial x^2} - \eta T + u(t),$$

where $T(x, t)$ denotes the temperature distribution at location $x$ and time $t$. The length of the slab $L = 0.6 m$. $K(x, T)$ denotes the thermal diffusivity, $\eta$ denotes the convective heat transfer coefficient, and $u(t)$ denotes the external heat sources.

The initial condition and boundary conditions are:

$$T(x, 0) = 100^\circ F,$$

$$\frac{\partial T}{\partial x}|_{x=0} = 0, T(L, t) = 150^\circ F.$$
The rank of the Hankel matrix $n_r = 20$, and hence, the identified reduced order system $A_k$ is in $\mathbb{R}^{20 \times 20}$. Due to the separation principle, the feedback design decouples into the solution of two $20 \times 20$ Ricatti equations, one for the controller and one for the Kalman filter: compare this to the 10100 x 10100 problem that would need to be solved if using an ILOG based approach. With the identified linearized system, we design the closed loop controller. We run 1000 individual simulations with process noise $w_k \sim N(0,W)$ and measurement noise $v_k \sim N(0,V)$, where $W = I_{5 \times 5}$, $V = I_{5 \times 5}$. In Fig. 2, we show the performance of the proposed approach. We calculate the identified Markov parameters using $(A_k, B_k, C_k)$, and compare with the actual generalized Markov parameters (calculated using impulse responses). The Markov parameters $h_{k,j} \in \mathbb{R}^{5 \times 5}$, and we show the comparison for one input-output channel at time step $k = 250$ in Fig 2(e). It can be seen that the identified LTV system using time-varying ERA approach can approximate the linearized deviation system accurately. In Fig. 2, we compare the averaged closed loop trajectory with the nominal trajectory at time $t = 37.5s$, $t = 62.5s$. In Fig. 2(e) - (f), we randomly choose two positions, and show the errors between the actual trajectory and optimal trajectory with 2σ bound in one simulation. For comparison, the open loop error is also shown in the figure. It can be seen that the averaged state estimates over 1000 Monte-Carlo simulations runs are close to the open loop optimal trajectory, which implies that the control objective to minimize the expected cost function could be achieved using the proposed approach. In this partially observed problem, the computationally complexity of designing the online estimator and controller using the identified ROM model are reduced by orders of $O(10^5)$, and for a general three dimensional problem this reduction could be even more significant.

V. CONCLUSION

In this paper, we have proposed a separation based design of the stochastic optimal control problem for systems with unknown nonlinear dynamics and partially observed states. First, we design a deterministic open-loop optimal trajectory in belief space. Then we identify the nominal linearized system using time-varying ERA. The open-loop optimization and system identification are implemented offline, using the impulse responses of the system, and an LQG controller based on the ROM is implemented online. The offline learning procedure is simple, and the online implementation is fast. We have tested the proposed approach on a one dimensional nonlinear heat transfer problem, and showed the performance of the proposed approach.

APPENDIX A

ENSEMBLE KALMAN FILTER

Consider the discrete time nonlinear dynamical system with Gaussian belief states $b_k = (\mu_k, \Sigma_k)$. Assume that $b_0$ is known. Denote $\{x_k^{(j)}\}$, $j = 1, \cdots, m$ as an $m$-member forecast ensemble at time step $k$, where the subscript $(j)$ denotes the $j^{th}$ member, the forecast mean $\hat{b}_k$ and covariance $\Sigma_k$ are defined as

$$b_k^- = \frac{1}{m} \sum_{j=1}^{m} x_k^{(j)},$$

$$\Sigma_k^- = \frac{1}{m-1} \sum_{j=1}^{m} (x_k^{(j)} - b_k^-) (x_k^{(j)} - b_k^-)'.$$  \hspace{1cm} (29)

The measurement ensemble at time $k$ is $\{y_k^{(j)}\}$, $j = 1, \cdots, m$, where $y_k^{(j)} = h(x_k^{(j)}, v_k), v_k \sim N(0, V)$. The corresponding mean and covariance $y_k, P^y_k$ are defined similarly.

Denote the cross-covariance matrix $P^xy_k$ between the state and measurement ensembles at time $k$ as:

$$P^xy_k = \sum_{j=1}^{m} (x_k^{(j)} - b_k^-) (y_k^{(j)} - y_k^-)'.$$  \hspace{1cm} (31)

Given an initial belief $b_0$, and a control sequence $\{u_k\}_{k=0}^{N-1}$, the EnKF used to estimate the belief state $\{b_k\}_{k=0}^{N}$ is summarized in Algorithm 4.

**Algorithm 4 EnKF**

**Require:** Start belief $b_0$, control sequence $\{u_k\}_{k=0}^{N-1}$.
**Ensure:** Belief nominal trajectory $\{b_k\}_{k=0}^{N}$.

1: for $k = 0 : N$ do
2: Sample $m$ forecast ensemble members:
3: $x_{k+1, (j)} = f(b_k, u_k, w_k)$
4: $y_{k+1, (j)} = h(x_{k+1, (j)}, v_{k+1})$, $j = 1, \cdots, m$ \hspace{1cm} (32)
5: where $w_k \sim (0, W)$, $v_k \sim (0, V)$, and calculate
6: $b_k^{+1}, \Sigma_k^{+1}, y_k^{+1}, P_{k+1}^x, P_{k+1}^y$.
7: Propagate the underlying noiseless system and take the measurement $y_k+1$:
8: $x_{k+1} = f(x_k, u_k, 0), y_{k+1} = h(x_{k+1}, 0)$.
9: Update the posterior mean and covariance
10: $b_{k+1} = b_k^{+1} + K_e (y_{k+1} - \hat{y}_{k+1})$
11: $\Sigma_{k+1} = \Sigma_k^{+1} - P_{k+1}^x (P_{k+1}^y)^{-1} P_{k+1}^x$, \hspace{1cm} (34)
12: where
13: $K_e = P_{k+1}^x (P_{k+1}^y)^{-1}$.
14: end for

APPENDIX B

CLOSED LOOP FEEDBACK CONTROLLER

Given the identified linear deviation system \cite{24}, the separation principle could be used, and hence, we design a Kalman filter and an LQR controller separately. The feedback controller is:

$$\delta u_k = -L_k \delta \hat{u}_k,$$  \hspace{1cm} (36)

where $\delta \hat{u}_k$ is the estimate from a Kalman observer, and the feedback gain $L_k$ is computed by solving two decoupled
Riccati equations as follows.

\[ L_k = (B'_k S_{k+1} B_k + R_k)^{-1} B'_k S_{k+1} \hat{A}_k, \]  

where \( S_k \) is determined by running the following Riccati equation backward in time:

\[ S_k = \hat{A}_k S_{k+1} \hat{A}_k^{T} + Q_k \]
\[ - A_k S_{k+1} \hat{B}_k (B'_k S_{k+1} \hat{B}_k + R_k)^{-1} B'_k S_{k+1} \hat{A}_k, \]

with terminal condition \( S_N = Q_N \).

The Kalman filter observer is designed as follows:

\[ \hat{\delta} a_{k+1} = \hat{A}_k \hat{\delta} a_k + \hat{B}_k \hat{\delta} u_k + K_{k+1}(\hat{C}_k \hat{\delta} a_k + \hat{B}_k \hat{\delta} u_k)), \]

with \( \hat{\delta} y_k = h(\bar{x}_k, v_k) - h(\bar{\mu}_k, 0) \), and the covariance of the estimation is:

\[ P_{k+1} = \hat{A}_k (P_k - P_k \hat{C}_k' (\hat{C}_k P_k \hat{C}_k' + V)^{-1} \hat{C}_k P_k) \hat{A}_k^{T} + \hat{B}_k W \hat{B}_k. \]

where the Kalman gain is:

\[ K_k = P_k \hat{C}_k' (\hat{C}_k P_k \hat{C}_k' + V)^{-1}. \]

**BIBLIOGRAPHY**

[1] D. P. Bertsekas, *Dynamic Programming and Optimal Control, Two Volume Set*, 2nd ed. Athena Scientific, 1995.

[2] M. Falcone, “Recent Results in the Approximation of Nonlinear Optimal Control Problems,” in *Large-Scale Scientific Computing LSSC*, 2013.

[3] A. Bryson and H. Y.-C., *Applied Optimal Control: Optimization, Estimation and Control*. Washington: Hemisphere Pub. Corp., 1975.

[4] D. Jacobsen and D. Mayne, *Differential Dynamic Programming*. Elsevier, 1970.

[5] E. Theodoro, Y. Tassa, and E. Todorov, “Stochastic Differential Dynamic Programming.” in *Proceedings of American Control Conference*, 2010.

[6] E. Todorov and W. Li, “A generalized iterative LQG method for locally-optimal feedback control of constrained nonlinear stochastic systems,” in *Proceedings of American Control Conference*, 2005, pp. 300 – 306.

[7] W. Li and E. Todorov, “Iterative linearization methods for approximately optimal control and estimation of non-linear stochastic system,” *International Journal of Control*, vol. 80, no. 9, pp. 1439–1453, 2007.

[8] R. P. Bithmead, V. Wertz, and M. Gerers, *Adaptive Optimal Control: The Thinking Man’s G.P.C*. Prentice Hall Professional Technical Reference, 1991.

[9] X. Zhong, H. He, H. Zhang, and Z. Wang, “Optimal Control for Unknown Discrete-Time Nonlinear Markov Jump Systems Using Adaptive Dynamic Programming,” *IEEE Transactions on Neural networks and learning systems*, vol. 25, no. 12, pp. 2141–2155, 2014.

[10] S. G. Khan et al., “Reinforcement learning and optimal adaptive control: An overview and implementation examples,” *Annual Reviews in Control*, vol. 36, pp. 42–59, 2012.

[11] D. Mitrovic, S. Klanke, and S. Vijayakumar, *Adaptive Optimal Feedback Control with Learned Internal Dynamics Models*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 65–84.

[12] S. Levine and P. Abbeel, “Learning Neural Network..."
Policies with Guided Search under Unknown Dynamics,” in Advances in Neural Information Processing Systems, 2014.

[13] S. Levine and K. Vladlen, “Learning Complex Neural Network Policies with Trajectory Optimization,” in Proceedings of the International Conference on Machine Learning, 2014.

[14] R. Akrour, A. Abdolmaleki, H. Abdulsamad, and G. Neumann, “Model-Free Trajectory Optimization for Reinforcement Learning,” in Proceedings of the International Conference on Machine Learning, 2016.

[15] E. Todorov and Y. Tassa, “Iterative Local Dynamic Programming,” in Proc. of the IEEE Int. Symposium on ADP and RL., 2009.

[16] J. Baxter and P. Bartlett, “Infinite Horizon Policy-Gradient Estimation,” Journal of Artificial Intelligence Research, vol. 15, pp. 319–350, 2001.

[17] R. S. Sutton, D. Mcallester, S. Singh, and Y. Mansour, “Policy Gradient Methods for Reinforcement Learning with Function Approximation,” in Proc. 1999 Neural Information Proc. Sys., 1999.

[18] M. P. Deisenroth, G. Neumann, and J. Peters, “A Survey on Policy Search for Robotics,” in Foundations and Trends in Robotics, 2013, pp. 1–142.

[19] M. Majji, J.-N. Juang, and J. L. Junkins, “Time-varying Eigensystem Realization Algorithm,” Journal of Guidance, Control, and Dynamics, vol. 33, no. 1, pp. 13–28, 2010.

[20] W. Favoreel et al., “Closed-loop model-free subspace-based lqg-design,” in Proceedings of the 7th Mediterranean Conference on Control and Automation, 1999, pp. 1926–1939.

[21] Z.-S. Hou and Z. Wang, “From model-based control to data-driven control: Survey, classification and perspective,” Information Sciences, vol. 235, no. 3-35, 2013.

[22] F. Hamilton, T. Berry, and T. Sauer, “Ensemble Kalman Filtering without a Model,” Physical Review X, vol. 6, p. 011021, 2016.

[23] A.E.Bryson and W. Denham, “A steepest-ascent method for solving optimum programming problems,” Journal of Applied Mechanics, vol. 29, no. 2, 1962.