Memory-Limited Partially Observable Stochastic Control and Its Mean-Field Control Approach

Takehiro Tottori 1, * and Tetsuya J. Kobayashi 1,2,3,4

1 Department of Mathematical Informatics, Graduate School of Information Science and Technology, The University of Tokyo, Tokyo 113-8654, Japan
2 Institute of Industrial Science, The University of Tokyo, Tokyo 153-8505, Japan
3 Department of Electrical Engineering and Information Systems, Graduate School of Engineering, The University of Tokyo, Tokyo 113-8654, Japan
4 Universal Biology Institute, The University of Tokyo, Tokyo 113-8654, Japan

* Correspondence: takehiro_tottori@sat.t.u-tokyo.ac.jp

Abstract: Control problems with incomplete information and memory limitation appear in many practical situations. Although partially observable stochastic control (POSC) is a conventional theoretical framework that considers the optimal control problem with incomplete information, it cannot consider memory limitation. Furthermore, POSC cannot be solved in practice except in special cases. In order to address these issues, we propose an alternative theoretical framework, memory-limited POSC (ML-POSC). ML-POSC directly considers memory limitation as well as incomplete information, and it can be solved in practice by employing the technique of mean-field control theory. ML-POSC can generalize the linear-quadratic-Gaussian (LQG) problem to include memory limitation. Because estimation and control are not clearly separated in the LQG problem with memory limitation, the Riccati equation is modified to the partially observable Riccati equation, which improves estimation as well as control. Furthermore, we demonstrate the effectiveness of ML-POSC for a non-LQG problem by comparing it with the local LQG approximation.

Keywords: decision-making; optimal control; stochastic control; incomplete information; memory limitation; mean-field control

1. Introduction

Control problems of systems with incomplete information and memory limitation appear in many practical situations. These constraints become especially predominant when designing the control of small devices [1,2], and are important for understanding the control mechanisms of biological systems [3–8] because their sensors are extremely noisy and their controllers can only have severely limited memories.

Partially observable stochastic control (POSC) is a conventional theoretical framework that considers the optimal control problem with one of these constraints, namely, the incomplete information of the system state (Figure 1b) [9]. Because the POSC controller cannot completely observe the state of the system, it determines the control based on the noisy observation history of the state. POSC can be solved in principle [10–12] by converting it to a completely observable stochastic control (COSC) of the posterior probability of the state, as the posterior probability represents the sufficient statistics of the observation history. The posterior probability and the optimal control are obtained by solving the Zakai equation and the Bellman equation, respectively.

However, POSC has three practical problems with respect to the implementation of the controller which originate from the ignorance of the other constraint, namely, the memory limitation of the controller [1,2]. First, a controller designed by POSC should ideally have an infinite-dimensional memory to store and compute the posterior probability from the observation history. Second, the memory of the controller cannot have intrinsic stochasticity
other than the observation noise to accurately compute the posterior probability via the Zakai equation. Third, POSC does not consider the cost originating from the memory update, which can be regarded as a cost of estimation. In light of the dualistic roles played by estimation and control, considering only control cost by ignoring estimation cost is asymmetric. As a result, POSC is not practical for control problems where the memory size, noise, and cost are non-negligible. Therefore, we need an alternative theoretical framework considering memory limitation to circumvent these three problems.

Furthermore, POSC has another crucial problem in obtaining the optimal state control by solving the Bellman equation [3,4]. Because the posterior probability of the state is infinite-dimensional, POSC corresponds to an infinite-dimensional COSC. In the infinite-dimensional COSC, the Bellman equation becomes a functional differential equation, which needs to be solved in order to obtain the optimal state control. However, solving a functional differential equation is generally intractable, even numerically.

In this work, we propose an alternative theoretical framework to the conventional POSC which can address the above-mentioned two issues. We call it memory-limited POSC (ML-POSC), in which memory limitation as well as incomplete information are directly accounted (Figure 1c). The conventional POSC derives the Zakai equation without considering memory limitations. Then, the optimal state control is supposed to be derived by solving the Bellman equation, even though we do not have any practical way to do this. In contrast, ML-POSC first postulates the finite-dimensional and stochastic memory dynamics explicitly by taking the memory limitation into account and then jointly optimizes the memory dynamics and state control by considering the memory and control costs. As a result, unlike the conventional POSC, ML-POSC finds both the optimal state control and the optimal memory dynamics with given memory limitations. Furthermore, we show that the Bellman equation of ML-POSC can be reduced to the Hamilton–Jacobi–Bellman (HJB) equation by employing a trick from the mean-field control theory [13–15]. While the Bellman equation is a functional differential equation, the HJB equation is a partial differential equation. As a result, ML-POSC can be solved, at least numerically.

The idea behind ML-POSC is closely related to that of the finite-state controller [16–22]. Finite-state controllers have been studied using the partially observable Markov decision process (POMDP), that is, the discrete time and state POSC. The finite-dimensional memory of ML-POSC can be regarded as an extension of the finite-state controller of POMDP to the continuous time and state setting. Nonetheless, the algorithms of the finite-state controller cannot be directly extended to the continuous setting, as they strongly depend on the discreteness. Although Fox and Tishby extended the finite-state controller to the continuous setting, their algorithm is restricted to the special case [1,2]. ML-POSC resolves this problem by employing the technique of the mean-field control theory.

In the linear-quadratic-Gaussian (LQG) problem of the conventional POSC, the Zakai equation and the Bellman equation are reduced to the Kalman filter and the Riccati equation, respectively [9,23]. Because the infinite-dimensional Zakai equation is reduced to the finite-dimensional Kalman filter, the LQG problem of the conventional POSC can be discussed in terms of ML-POSC. We show that the Kalman filter corresponds to the optimal memory dynamics of ML-POSC. Moreover, ML-POSC can generalize the LQG problem to include memory limitations such as the memory noise and cost. Because estimation and control are not clearly separated in the LQG problem with memory limitation, the Riccati equation for control is modified to include estimation, which in this paper is called the partially observable Riccati equation. We demonstrate that the partially observable Riccati equation is superior to the conventional Riccati equation as concerns the LQG problem with memory limitation.

Then, we investigate the potential effectiveness of ML-POSC for a non-LQG problem by comparing it with the local LQG approximation of the conventional POSC [3,4]. In the local LQG approximation, the Zakai equation and the Bellman equation are locally approximated by the Kalman filter and the Riccati equation, respectively. Because the Bellman equation (a functional differential equation) is reduced to the Riccati equation (an
ordinary differential equation), the local LQG approximation can be solved numerically. However, the performance of the local LQG approximation may be poor in a highly non-LQG problem, as the local LQG approximation ignores non-LQG information. In contrast, ML-POSC reduces the Bellman equation to the HJB equation while maintaining non-LQG information. We demonstrate that ML-POSC can provide a better result than the local LQG approximation in a non-LQG problem.

This paper is organized as follows: In Section 2, we briefly review the conventional POSC. In Section 3, we formulate ML-POSC. In Section 4, we propose the mean-field control approach to ML-POSC. In Section 5, we investigate the LQG problem of the conventional POSC based on ML-POSC. In Section 6, we generalize the LQG problem to include memory limitation. In Section 7, we show numerical experiments involving a LQG problem with memory limitation and a non-LQG problem. Finally, in Section 8, we discuss our work.

![Schematic diagram of (a) completely observable stochastic control (COSC), (b) partially observable stochastic control (POSC), and (c) memory-limited partially observable stochastic control (ML-POSC).](image)

**Figure 1.** Schematic diagram of (a) completely observable stochastic control (COSC), (b) partially observable stochastic control (POSC), and (c) memory-limited partially observable stochastic control (ML-POSC). The top and bottom figures represent the system and controller, respectively; \( x_t \in \mathbb{R}^d \) is the state of the system; \( y_t \in \mathbb{R}^d, z_t \in \mathbb{R}^d \), and \( u_t \in \mathbb{R}^d \) are the observation, memory, and control of the controller, respectively. (a) In COSC, the controller can completely observe the state \( x_t \), and determines the control \( u_t \) based on the state \( x_t \), i.e., \( u_t = u(t, x_t) \). Only finite-dimensional memory is required to store the state \( x_t \), and the optimal control \( u_t^* \) is obtained by solving the Hamilton–Jacobi–Bellman (HJB) equation, which is a partial differential equation. (b) In POSC, the controller cannot completely observe the state \( x_t \); instead, it obtains the noisy observation \( y_t \) of the state \( x_t \). The control \( u_t \) is determined based on the observation history \( y_{0:t} := \{y_\tau | \tau \in [0, t] \} \), i.e., \( u_t = u(t, y_{0:t}) \). An infinite-dimensional memory is implicitly assumed to store the observation history \( y_{0:t} \). Furthermore, to obtain the optimal control \( u_t^* \), the Bellman equation (a functional differential equation) needs to be solved, which is generally intractable, even numerically. (c) In ML-POSC, the controller is only accessible to the noisy observation \( y_t \) of the state \( x_t \), as in POSC. In addition, it has only finite-dimensional memory \( z_t \), which cannot completely memorize the observation history \( y_{0:t} \). The controller of ML-POSC compresses the observation history \( y_{0:t} \) into the finite-dimensional memory \( z_t \), then determines the control \( u_t \) based on the memory \( z_t \), i.e., \( u_t = u(t, z_t) \). The optimal control \( u_t^* \) is obtained by solving the HJB equation (a partial differential equation), as in COSC.

2. Review of Partially Observable Stochastic Control

In this section, we briefly review the conventional POSC [11,15].
2.1. Problem Formulation

In this subsection, we formulate the conventional POSC [11,15]. The state \( x_t \in \mathbb{R}^d \) and the observation \( y_t \in \mathbb{R}^d \) at time \( t \in [0,T] \) evolve by the following stochastic differential equations (SDEs):

\[
\begin{align*}
    dx_t &= b(t, x_t, u_t)dt + \sigma(t, x_t, u_t)d\omega_t, \\
    dy_t &= h(t, x_t)dt + \gamma(t)dv_t,
\end{align*}
\]

(1)

(2)

where \( x_0 \) and \( y_0 \) obey \( p_0(x_0) \) and \( p_0(y_0) \), respectively, \( \omega_t \in \mathbb{R}^d \) and \( u_t \in \mathbb{R}^d \) are independent standard Wiener processes, and \( u_t \) is the control. Here, \( \gamma(t)\gamma^\top(t) \) is assumed to be invertible. In POSC, because the controller cannot completely observe the state \( x_t \), the control \( u_t \) is determined based on the observation history \( y_{0:t} := \{ y_t | t \in [0,t] \} \), as follows:

\[
u_t = u(t, y_{0:t}).
\]

(3)

The objective function of POSC is provided by the following expected cumulative cost function:

\[
J[u] := \mathbb{E}_{p(x_{0:T}, y_{0:T}; u)} \left[ \int_0^T f(t, x_t, u_t)dt + g(x_T) \right],
\]

(4)

where \( f \) is the cost function, \( g \) is the terminal cost function, \( p(x_{0:T}, y_{0:T}; u) \) is the probability of \( x_{0:T} \) and \( y_{0:T} \) given \( u \) as a parameter, and \( \mathbb{E}_{p[\cdot]} \) is the expectation with respect to probability \( p \). Throughout this paper, the time horizon \( T \) is assumed to be finite.

POSC is the problem of finding the optimal control function \( u^* \) that minimizes the objective function \( J[u] \) as follows:

\[
u^* := \arg\min_u J[u].
\]

(5)

2.2. Derivation of Optimal Control Function

In this subsection, we briefly review the derivation of the optimal control function of the conventional POSC [11,15]. We first define the unnormalized posterior probability density function \( q_t(x) := p(x_t = x, y_{0:t}) \). We omit \( y_{0:t} \) for notational simplicity. Here, \( q_t(x) \) obeys the following Zakai equation:

\[
dq_t(x) = \mathcal{L}^\dagger q_t(x)dt + q_t(x)h^\top(t, x)(\gamma(t)\gamma^\top(t))^{-1}dy_t,
\]

(6)

where \( q_0(x) = p_0(x)p_0(y) \) and \( \mathcal{L}^\dagger \) is the forward diffusion operator, which is defined by

\[
\mathcal{L}^\dagger q(x) := -\sum_{i=1}^d \frac{\partial (b_i(t, x, u)q(x))}{\partial x_i} + \frac{1}{2} \sum_{i,j=1}^d \frac{\partial^2 (D_{ij}(t, x, u)q(x))}{\partial x_i \partial x_j},
\]

(7)

where \( D(t, x, u) := \sigma(t, x, u)\sigma^\top(t, x, u) \). Then, the objective function \( J[u] \) can be calculated as follows:

\[
J[u] = \mathbb{E}_{p(\phi_{0:T}; u)} \left[ \int_0^T \tilde{f}(t, q_t, u_t)dt + \tilde{g}(q_T) \right],
\]

(8)

where \( \tilde{f}(t, q, u) := \mathbb{E}_{q_t(x)}[f(t, x, u)] \) and \( \tilde{g}(q) := \mathbb{E}_{q(x)}[g(x)] \). From (6) and (8), POSC is converted into a COSC of \( q_t \). As a result, POSC can be approached in the similar way as COSC, and the optimal control function is provided by the following proposition.
Proposition 1 ([11,15]). The optimal control function of POSC is provided by

\[
\begin{align*}
    u^*(t,q) = \arg\min_u \mathbb{E}_{q(x)} \left[ \mathcal{H} \left( t, x, u, \frac{\delta V(t,q)}{\delta q} (x) \right) \right],
\end{align*}
\]

where \( \mathcal{H} \) is the Hamiltonian, which is defined by

\[
\begin{align*}
    \mathcal{H} \left( t, x, u, \frac{\delta V(t,q)}{\delta q} (x) \right) := f(t,x,u) + \mathcal{L} \frac{\delta V(t,q)}{\delta q} (x).
\end{align*}
\]

\( \mathcal{L} \) is the backward diffusion operator, which is defined by

\[
\begin{align*}
    \mathcal{L} q(x) := \sum_{i=1}^{d_x} b_i(t,x,u) \frac{\partial q(x)}{\partial x_i} + \frac{1}{2} \sum_{ij=1}^{d_x} D_{ij}(t,x,u) \frac{\partial^2 q(x)}{\partial x_i \partial x_j}.
\end{align*}
\]

We note that \( \mathcal{L} \) is the conjugate of \( \mathcal{L}^\dagger \); furthermore, \( V(t,q) \) is the value function, which is the solution of the following Bellman equation:

\[
\begin{align*}
    -\frac{\partial V(t,q)}{\partial t} &= \mathbb{E}_{q(x)} \left[ \mathcal{H} \left( t, x, u^*, \frac{\delta V(t,q)}{\delta q} (x) \right) \right] \\
    &+ \frac{1}{2} \mathbb{E}_{q(x,q(x'))} \left[ \frac{\delta V(t,q)}{\delta q} (x,x') \right] h^\top (t,x) (\gamma(t)\gamma^\top(t))^{-1} h(t,x'),
\end{align*}
\]

where \( V(T,q) = \mathbb{E}_{q(x)} [g(x)] \).

**Proof.** The proof is shown in [11,15]. \( \square \)

The optimal control function \( u^*(t,q) \) is obtained by solving the Bellman Equation (12). The controller determines the optimal control \( u^*_i = u^*(t,q_i) \) based on the posterior probability \( q_i \). The posterior probability \( q_i \) is obtained by solving the Zakai Equation (6). As a result, POSC can be solved in principle.

However, POSC has three practical problems with respect to the memory of the controller. First, the controller should have an infinite-dimensional memory to store and compute the posterior probability \( q_i \) from the observation history \( y_{[0,t]} \). Second, the memory of the controller cannot have intrinsic stochasticity other than the observation \( dy_t \) to accurately compute the posterior probability \( q_i \) via the Zakai Equation (6). Third, POSC does not consider the cost originating from the memory update, which can be regarded as a cost of estimation. In light of the dualistic roles played by estimation and control, considering only control cost by ignoring estimation cost is asymmetric. As a result, POSC is not practical for control problems where the memory size, noise, and cost are non-negligible. Furthermore, POSC has another crucial problem in obtaining the optimal control function \( u^*(t,q) \) by solving the Bellman Equation (12). Because the posterior probability \( q \) is infinite-dimensional, the associated Bellman Equation (12) becomes a functional differential equation. However, solving a functional differential equation is generally intractable even numerically. As a result, POSC cannot be solved in practice.

3. Memory-Limited Partially Observable Stochastic Control

In order to address the above-mentioned problems, we propose an alternative theoretical framework to the conventional POSC called ML-POSC. In this section, we formulate ML-POSC.
3.1. Problem Formulation

In this subsection, we formulate ML-POSC. ML-POSC determines the control $u_t$ based on the finite-dimensional memory $z_t \in \mathbb{R}^{d_z}$ as follows:

$$u_t = u(t, z_t).$$  \hspace{1cm} (13)

The memory dimension $d_z$ is determined not by the optimization but by the prescribed memory limitation of the controller to be used. Comparing (3) and (13), the memory $z_t$ can be interpreted as the compression of the observation history $y_{0:t}$. While the conventional POSC compresses the observation history $y_{0:t}$ into the infinite-dimensional posterior probability $q_t$, ML-POSC compresses it into the finite-dimensional memory $z_t$.

ML-POSC formulates the memory dynamics with the following SDE:

$$dz_t = c(t, z_t, v_t)dt + \kappa(t, z_t, v_t)dy_t + \eta(t, z_t, v_t)d\xi_t,$$  \hspace{1cm} (14)

where $z_0$ obeys $p_0(z_0)$, $\xi_t \in \mathbb{R}^{d_\xi}$ is the standard Wiener process, and $v_t = v(t, z_t) \in \mathbb{R}^{d_v}$ is the control for the memory dynamics. This memory dynamics has three important properties: (i) because it depends on the observation $dy_t$, the memory $z_t$ can be interpreted as the compression of the observation history $y_{0:t}$; (ii) because it depends on the standard Wiener process $d\xi_t$, ML-POSC can consider the memory noise explicitly; (iii) because it depends on the control $v_t$, it can be optimized through the control $v_t$.

The objective function of ML-POSC is provided by the following expected cumulative cost function:

$$J[u, v] := \mathbb{E}_{p(x_{0:T}, y_{0:T}, z_{0:T}, u, v)} \left[ \int_0^T f(t, x_t, u_t, v_t)dt + g(x_T) \right].$$  \hspace{1cm} (15)

Because the cost function $f$ depends on the memory control $v_t$ as well as the state control $u_t$, ML-POSC can consider the memory control cost (state estimation cost) as well as the state control cost explicitly.

ML-POSC optimizes the state control function $u$ and the memory control function $v$ based on the objective function $J[u, v]$, as follows:

$$u^*, v^* := \arg\min_{u, v} J[u, v].$$  \hspace{1cm} (16)

ML-POSC first postulates the finite-dimensional and stochastic memory dynamics explicitly, then jointly optimizes the state and memory control function by considering the state and memory control cost. As a result, unlike the conventional POSC, ML-POSC can consider memory limitation as well as incomplete information.

3.2. Problem Reformulation

Although the formulation of ML-POSC in the previous subsection clarifies its relationship with that of the conventional POSC, it is inconvenient for further mathematical investigations. In order to resolve this problem, we reformulate ML-POSC in this subsection. The formulation in this subsection is simpler and more general than that in the previous subsection.

We first define the extended state $s_t$ as follows:

$$s_t := \begin{pmatrix} x_t \\ z_t \end{pmatrix} \in \mathbb{R}^{d_s},$$  \hspace{1cm} (17)

where $d_s = d_x + d_z$. The extended state $s_t$ evolves by the following SDE:

$$ds_t = \bar{b}(t, s_t, \bar{u}_t)dt + \bar{\sigma}(t, s_t, \bar{u}_t)d\bar{\omega}_t,$$  \hspace{1cm} (18)
where $s_0$ obeys $p_0(s_0), \tilde{\omega}_t \in \mathbb{R}^{d_\omega}$ is the standard Wiener process, and $\tilde{u}_t \in \mathbb{R}^{d_u}$ is the control. ML-POSC determines the control $\tilde{u}_t \in \mathbb{R}^{d_u}$ based solely on the memory $z_t$, as follows:

$$\tilde{u}_t = \tilde{u}(t, z_t). \quad (19)$$

The extended state SDE (18) includes the previous state, observation, and memory SDEs (1), (2) and (14) as a special case; they can be represented as follows:

$$d s_t = \left( b(t, x_t, u_t) \right) dt + \left( \sigma(t, x_t, u_t) \right) \left( \frac{d \omega_t}{d t} \right), \quad (20)$$

where $p_0(s_0) = p_0(x_0)p_0(z_0)$.

The objective function of ML-POSC is provided by the following expected cumulative cost function:

$$J[\tilde{u}] := E_{p(s_0:T; \tilde{u})} \left[ \int_0^T \tilde{f}(t, s_t, \tilde{u}_t) dt + \tilde{g}(s_T) \right], \quad (21)$$

where $\tilde{f}$ is the cost function and $\tilde{g}$ is the terminal cost function. It is obvious that this objective function (21) is more general than the previous one (15).

ML-POSC is the problem of finding the optimal control function $\tilde{u}^*$ that minimizes the objective function $J[\tilde{u}]$ as follows:

$$\tilde{u}^* := \arg\min_{\tilde{u}} J[\tilde{u}]. \quad (22)$$

In the following section, we mainly consider the formulation in this subsection rather than that of the previous subsection, as it is simpler and more general. Moreover, we omit $\tilde{\cdot}$ for the notational simplicity.

4. Mean-Field Control Approach

If the control $u_t$ is determined based on the extended state $s_t$, i.e., $u_t = u(t, s_t)$, ML-POSC is the same as COSC of the extended state $s_t$, and can be solved by the conventional COSC approach [10]. However, because ML-POSC determines the control $u_t$ based solely on the memory $z_t$, i.e., $u_t = u(t, z_t)$, ML-POSC cannot be solved in a similar way as COSC. In order to solve ML-POSC, we propose the mean-field control approach in this section. Because the mean-field control approach is more general than the COSC approach, it can solve COSC and ML-POSC in a unified way.

4.1. Derivation of Optimal Control Function

In this subsection, we propose the mean-field control approach to ML-POSC. We first show that ML-POSC can be converted into a deterministic control of the probability density function, which is similar to the conventional POSC [11,15]. This approach is used in the mean-field control as well [13,14,24,25]. The extended state SDE (18) can be converted into the following Fokker–Planck (FP) equation:

$$\frac{d p_t(s)}{d t} = \mathcal{L}^t p_t(s), \quad (23)$$

where the initial condition is provided by $p_0(s)$ and the forward diffusion operator $\mathcal{L}^t$ is defined by (7). The objective function of ML-POSC (21) can be calculated as follows:

$$J[u] = \int_0^T \tilde{f}(t, p_t, u_t) dt + \tilde{g}(p_T), \quad (24)$$
where \( \bar{f}(t, p, u) := \mathbb{E}_{p(t)}[f(t, s, u)] \) and \( \bar{g}(p) := \mathbb{E}_{p(s)}[g(s)] \). From (23) and (24), ML-POSC is converted into a deterministic control of \( p_t \). As a result, ML-POSC can be approached in a similar way as the deterministic control, and the optimal control function is provided by the following lemma.

**Lemma 1.** The optimal control function of ML-POSC is provided by

\[
u^*(t, z) = \arg\min_u \mathbb{E}_{p_t(z)} \left[ \mathcal{H} \left( t, s, u, \frac{\partial V(t, p_t)}{\partial p} (s) \right) \right],
\]

where \( \mathcal{H} \) is the Hamiltonian (10), \( p_t(x|z) = p_t(s) / \int p_t(s)dx \) is the conditional probability density function of a state \( x \) given memory \( z \), \( p_t(s) \) is the solution of the FP Equation (23), and \( V(t, p) \) is the solution of the following Bellman equation:

\[
- \frac{\partial V(t, p)}{\partial t} = \mathbb{E}_{p(s)} \left[ \mathcal{H} \left( t, s, u^*, \frac{\partial V(t, p)}{\partial p} (s) \right) \right],
\]

where \( V(T, p) = \mathbb{E}_{p(s)}[g(s)] \).

**Proof.** The proof is shown in Appendix A. \( \square \)

The controller of ML-POSC determines the optimal control \( u^*_t = u^*(t, z_t) \) based on the memory \( z_t \), not the posterior probability \( q_t \). Therefore, ML-POSC can consider memory limitation as well as incomplete information.

However, because the Bellman Equation (26) is a functional differential equation, it cannot be solved, even numerically, which is the same problem as the conventional POSC. We resolve this problem by employing the technique of the mean-field control theory [13,14] as follows.

**Theorem 1.** The optimal control function of ML-POSC is provided by

\[
u^*(t, z) = \arg\min_u \mathbb{E}_{p_t(z)} \left[ \mathcal{H} \left( t, s, u, \bar{w}(t, s) \right) \right],
\]

where \( \mathcal{H} \) is the Hamiltonian (10), \( p_t(x|z) = p_t(s) / \int p_t(s)dx \) is the conditional probability density function of a state \( x \) given memory \( z \), \( p_t(s) \) is the solution of the FP Equation (23), and \( \bar{w}(t, s) \) is the solution of the following Hamilton–Jacobi–Bellman (HJB) equation:

\[
- \frac{\partial \bar{w}(t, s)}{\partial t} = \mathcal{H} \left( t, s, u^*, \bar{w}(t, s) \right),
\]

where \( \bar{w}(T, s) = g(s) \).

**Proof.** The proof is shown in Appendix B. \( \square \)

While the Bellman Equation (26) is a functional differential equation, the HJB Equation (28) is a partial differential equation. As a result, unlike the conventional POSC, ML-POSC can be solved in practice.

We note that the mean-field control technique is applicable to the conventional POSC as well, and we obtain the HJB equation of the conventional POSC [15]. However, the HJB equation of the conventional POSC is not closed by a partial differential equation due to the last term of the Bellman Equation (12). As a result, the mean-field control technique is not effective with the conventional POSC except in a special case [15].

In the conventional POSC, the state estimation (memory control) and the state control are clearly separated. As a result, the state estimation and the state control are optimized by the Zakai Equation (6) and the Bellman Equation (12), respectively. In contrast, because ML-POSC considers memory limitation as well as incomplete information, the state estimation and the state control are not clearly separated. As a result, ML-POSC jointly optimizes
the state estimation and the state control based on the FP Equation (23) and the HJB Equation (28).

4.2. Comparison with Completely Observable Stochastic Control

In this subsection, we show the similarities and differences between ML-POSC and COSC of the extended state. While ML-POSC determines the control \( u_t \) based solely on the memory \( z_t \), i.e., \( u_t = u(t, z_t) \), COSC of the extended state determines the control \( u_t \) based on the extended state \( s_t \), i.e., \( u_t = u(t, s_t) \). The optimal control function of COSC of the extended state is provided by the following proposition.

**Proposition 2 ([10]).** The optimal control function of COSC of the extended state is provided by

\[
u^*(t, s) = \arg\min_u \mathcal{H}(t, s, u, w(t, s)),
\]

where \( \mathcal{H} \) is the Hamiltonian (10) and \( w(t, s) \) is the solution of the HJB Equation (28).

**Proof.** The conventional proof is shown in [10]. We note that it can be proven in a similar way as ML-POSC, which is shown in Appendix C.

Although the HJB Equation (28) is the same between ML-POSC and COSC, the optimal control function is different. While the optimal control function of COSC is provided by the minimization of the Hamiltonian (29), that of ML-POSC is provided by the minimization of the conditional expectation of the Hamiltonian (27). This is reasonable, as the controller of ML-POSC needs to estimate the state from the memory.

4.3. Numerical Algorithm

In this subsection, we briefly explain a numerical algorithm to obtain the optimal control function of ML-POSC (27). Because the optimal control function of COSC (29) depends only on the backward HJB Equation (28), it can be obtained by solving the HJB equation backwards from the terminal condition [10,26,27]. In contrast, because the optimal control function of ML-POSC (27) depends on the forward FP Equation (23) as well as the backward HJB Equation (28), it cannot be obtained in a similar way as COSC. Because the backward HJB equation depends on the forward FP equation through the optimal control function of ML-POSC, the HJB equation cannot be solved backwards from the terminal condition. As a result, ML-POSC needs to solve the system of HJB-FP equations.

The system of HJB-FP equations appears in the mean-field game and control [28–30], and many numerical algorithms have been developed [31–33]. Therefore, unlike the conventional POSC, ML-POSC can be solved in practice using these algorithms. Furthermore, unlike the mean-field game and control, the coupling of HJB-FP equations is limited to the optimal control function in ML-POSC. By exploiting this property, more efficient algorithms may be proposed for ML-POSC [34].

In this paper, we use the forward–backward sweep method (the fixed-point iteration method) to obtain the optimal control function of ML-POSC [33–37], which is one of the most basic algorithms for the system of HJB-FP equations. The forward–backward sweep method computes the forward FP Equation (23) and the backward HJB Equation (28) alternately. In the mean-field game and control, the convergence of the forward–backward sweep method is not guaranteed. In contrast, it is guaranteed in ML-POSC because the coupling of HJB-FP equations is limited to the optimal control function [34].

5. Linear-Quadratic-Gaussian Problem without Memory Limitation

In the LQG problem of the conventional POSC, the Zakai Equation (6) and the Bellman Equation (12) are reduced to the Kalman filter and the Riccati equation, respectively [9,23]. Because the infinite-dimensional Zakai equation is reduced to the finite-dimensional Kalman filter, the LQG problem of the conventional POSC can be discussed in terms of ML-POSC. In this section, we briefly review the LQG problem of the conventional POSC,
then reproduce the Kalman filter and the Riccati equation from the viewpoint of ML-POSC. The LQG problem of the conventional POSC corresponds to the LQG problem without memory limitation, as it does not consider the memory noise and cost.

5.1. Review of Partially Observable Stochastic Control

In this subsection, we briefly review the LQG problem of the conventional POSC [9,23]. The state $x_t \in \mathbb{R}^{d_x}$ and the observation $y_t \in \mathbb{R}^{d_y}$ at time $t \in [0,T]$ evolve by the following SDEs:

\[
\begin{align*}
    dx_t &= (A(t)x_t + B(t)u_t)dt + \sigma(t)d\omega_t, \\
    dy_t &= H(t)x_tdt + \gamma(t)d\nu_t,
\end{align*}
\]

(30) (31)

where $x_0$ obeys the Gaussian distribution $p_0(x_0) = \mathcal{N}(x_0|\mu_0,\Sigma_{x0})$, $y_0$ is an arbitrary real vector, $\omega_t \in \mathbb{R}^{d_\omega}$ and $\nu_t \in \mathbb{R}^{d_\nu}$ are independent standard Wiener processes, and $u_t = u(t,y_{0:t}) \in \mathbb{R}^{d_u}$ is the control. Here, $\gamma(t)\gamma^T(t)$ is assumed to be invertible. The objective function is provided by the following expected cumulative cost function:

\[
J[u] := \mathbb{E}_{P_{x_0,T,y_0,T,u}} \left[ \int_0^T \left( x_t^TQ(t)x_t + u_t^TR(t)u_t \right)dt + x_T^TPx_T \right],
\]

(32)

where $Q(t) \succeq O$, $R(t) \succ O$, and $P \succeq O$. The LQG problem of the conventional POSC is to find the optimal control function $u^*$ that minimizes the objective function $J[u]$, as follows:

\[
u^* := \arg\min_u J[u].
\]

(33)

In the LQG problem of the conventional POSC, the posterior probability is provided by the Gaussian distribution $p(x_t|y_{0:t}) = \mathcal{N}(x_t|\hat{\mu}(t),\hat{\Sigma}(t))$, and $u_t = u(t,y_{0:t})$ is reduced to $u_t = u(t,\hat{\mu}_t)$ without loss of performance.

**Proposition 3** ([9,23]). In the LQG problem without memory limitation, the optimal control function of POSC (33) is provided by

\[
    u^*(t,\hat{\mu}) = -R^{-1}B^T\Psi\hat{\mu},
\]

(34)

where $\hat{\mu}(t)$ and $\hat{\Sigma}(t)$ are the solutions of the following Kalman filter:

\[
\begin{align*}
    d\hat{\mu} &= (A - BR^{-1}B^T\Psi)\hat{\mu}dt + \Sigma(\gamma\gamma^T)^{-1}(dy_t - H\hat{\mu}dt), \\
    \frac{d\hat{\Sigma}}{dt} &= \sigma\sigma^T + A\hat{\Sigma} + \hat{\Sigma}A^T - \hat{\Sigma}H(\gamma\gamma^T)^{-1}H^T\hat{\Sigma},
\end{align*}
\]

(35) (36)

and where $\hat{\mu}(0) = \mu_{x,0}$ and $\hat{\Sigma}(0) = \Sigma_{xx,0}$. $\Psi(t)$ is the solution of the following Riccati equation:

\[
    -\frac{d\Psi}{dt} = Q + A^T\Psi + \Psi A - \Psi BR^{-1}B^T\Psi,
\]

(37)

where $\Psi(T) = P$.

**Proof.** The proof is shown in [9,23]. □

In the LQG problem of the conventional POSC, the Zakai Equation (6) and the Bellman Equation (12) are reduced to the Kalman filter (35) and (36) and the Riccati Equation (37), respectively.

5.2. Memory-Limited Partially Observable Stochastic Control

Because the infinite-dimensional Zakai Equation (6) is reduced to the finite-dimensional Kalman filter (35) and (36), the LQG problem of the conventional POSC can be discussed in
We note that the LQG problem of the conventional POSC does not consider the memory.

Theorem 2. \( z_t \in \mathbb{R}^{d_z} \). In the LQG problem of the conventional POSC, the memory dimension \( d_z \) is the same as the state dimension \( d_x \). The controller of ML-POSC determines the control \( u_t \) based on the memory \( z_t \), i.e., \( u_t = u(t, z_t) \). The memory \( z_t \) is assumed to evolve by the following SDE:

\[
dz_t = v_t dt + \kappa_t dy_t,
\]

where \( z_0 = \mu_{0,zz} \), while \( v_t = v(t, z_t) \in \mathbb{R}^{d_z} \) and \( \kappa_t = \kappa(t, z_t) \in \mathbb{R}^{d_z \times d_y} \) are the memory controls. We note that the LQG problem of the conventional POSC does not consider the memory noise. The objective function of ML-POSC is provided by the following expected cumulative cost function:

\[
J[u, v, \kappa] := \mathbb{E}_{p(z_0, t \in T, y_0, T, z_0, t; u, v, \kappa)} \left[ \int_0^T \left( x_t^\top Q(t) x_t + u_t^\top R(t) u_t \right) dt + x_T^\top P x_T \right].
\] (39)

We note that the LQG problem of the conventional POSC does not consider the memory control cost. ML-POSC optimizes \( u, v, \) and \( \kappa \) based on \( J[u, v, \kappa] \), as follows:

\[
u^*, v^*, \kappa^* := \arg\min_{u, v, \kappa} J[u, v, \kappa].
\] (40)

In the LQG problem of the conventional POSC, the probability of the extended state \( s_t \) (17) is provided by the Gaussian distribution \( p_t(s_t) = \mathcal{N}(s_t|\mu(t), \Sigma(t)) \). The posterior probability of the state \( x_t \) given the memory \( z_t \) is provided by the Gaussian distribution \( p_t(x_t|z_t) = \mathcal{N}(x(t)|\mu_{x|z}(t, z_t), \Sigma_{x|z}(t)) \), where \( \mu_{x|z}(t, z_t) \) and \( \Sigma_{x|z}(t) \) are provided as follows:

\[
\mu_{x|z}(t, z_t) = \mu_{x}(t) + \Sigma_{xz}(t) \Sigma_{zz}^{-1}(t)(z_t - \mu_{z}(t)),
\] (41)

\[
\Sigma_{x|z}(t) = \Sigma_{xx}(t) - \Sigma_{xz}(t) \Sigma_{zz}^{-1}(t) \Sigma_{zx}(t).
\] (42)

Theorem 2. In the LQG problem without memory limitation, the optimal control functions of ML-POSC (40) are provided by

\[
u^*(t, z) = -R^{-1}B^\top \Psi z,
\] (43)

\[
v^*(t, z) = \left( A - BR^{-1}B^\top \Psi - \Sigma_{x|z}H^\top (\gamma \gamma^\top)^{-1}H \right) z,
\] (44)

\[
\kappa^*(t, z) = \Sigma_{z|z}H^\top (\gamma \gamma^\top)^{-1}.
\] (45)

From \( v^*(t, z) \) and \( \kappa^*(t, z) \), \( z_t \) and \( \Sigma_{x|z}(t) \) obey the following equations:

\[
dz_t = \left( A - BR^{-1}B^\top \Psi \right) z_t dt + \Sigma_{x|z}H^\top (\gamma \gamma^\top)^{-1}(dy_t - Hz_t dt),
\] (46)

\[
d\Sigma_{x|z} = \sigma \sigma^\top + A\Sigma_{x|z} + \Sigma_{x|z}A^\top - \Sigma_{x|z}H^\top (\gamma \gamma^\top)^{-1}H\Sigma_{x|z},
\] (47)

where \( z_0 = \mu_{x,0} \) and \( \Sigma_{x|z}(0) = \Sigma_{xx,0} \). Furthermore, \( \mu_{x|z}(t, z_t) = z_t \) holds in this problem. \( \Psi(t) \) is the solution of the Riccati Equation (37).

Proof. The proof is shown in Appendix D. \( \square \)

In the LQG problem of the conventional POSC, the optimal memory dynamics of ML-POSC (46) and (47) corresponds to the Kalman filter (35) and (36). Furthermore, ML-POSC reproduces the Riccati Equation (37).
6. Linear-Quadratic-Gaussian Problem with Memory Limitation

The LQG problem of the conventional POSC does not consider memory limitation because it does not consider the memory noise and cost. Furthermore, because the memory dimension is restricted to the state dimension, the memory dimension cannot be determined according to a given controller. ML-POSC can generalize the LQG problem to include the memory limitation. In this section, we discuss the LQG problem with memory limitation based on ML-POSC.

6.1. Problem Formulation

In this subsection, we formulate the LQG problem with memory limitation. The state and observation SDEs are the same as in the previous section, which are provided by (30) and (31), respectively. The controller of ML-POSC determines the control \( u_t \in \mathbb{R}^d_u \) based on the memory \( z_t \in \mathbb{R}^{d_z} \), i.e., \( u_t = u(t, z_t) \). Unlike the LQG problem of the conventional POSC, the memory dimension \( d_z \) is not necessarily the same as the state dimension \( d_x \).

The memory \( z_t \) is assumed to evolve according to the following SDE:

\[
dz_t = v_t dt + \kappa(t) dy_t + \eta(t) d\xi_t, \tag{48}
\]

where \( z_0 \) obeys the Gaussian distribution \( p_0(z_0) = N(z_0|\mu_z,0, \Sigma_{zz,0}) \), \( \xi_t \in \mathbb{R}^{d_\xi} \) is the standard Wiener process, and \( v_t = v(t, z_t) \in \mathbb{R}^{d_v} \) is the control. Because the initial condition \( z_0 \) is stochastic and the memory SDE (48) includes the intrinsic stochasticity \( d\xi_t \), the LQG problem of ML-POSC can consider the memory noise explicitly. We note that \( \kappa(t) \) is independent of the memory \( z_t \). If \( \kappa(t) \) depends on the memory \( z_t \), the memory SDE (48) becomes non-linear and non-Gaussian. As a result, the optimal control functions cannot be derived explicitly in this case. In order to keep the memory SDE (48) linear and Gaussian for obtaining the optimal control functions explicitly, we restrict \( \kappa(t) \) being independent of the memory \( z_t \) in the LQG problem with memory limitation. The LQG problem without memory limitation is the special case in which the optimal control \( \kappa^*_t = \kappa^*(t, z_t) \) in (45) does not depend on the memory \( z_t \).

The objective function is provided by the following expected cumulative cost function:

\[
J[u,v] := \mathbb{E}_p(x_0,T,y_0,T,z_0,T,u,v) \left[ \int_0^T \left( x_t^T Q(t) x_t + u_t^T R(t) u_t + v_t^T M(t) v_t \right) dt + x_T^T P x_T \right], \tag{49}
\]

where \( Q(t) \succ 0, R(t) \succ 0, M(t) \succ 0, \) and \( P \succeq 0 \). Because the cost function includes \( v_t^T M(t) v_t \), the LQG problem of ML-POSC can consider the memory control cost explicitly. ML-POSC optimizes the state control function \( u \) and the memory control function \( v \) based on the objective function \( J[u,v] \), as follows:

\[
u^*, u^* := \arg\min_{u,v} J[u,v]. \tag{50}
\]

For the sake of simplicity, we do not optimize \( \kappa(t) \), although this can be accomplished by considering unobservable stochastic control.

6.2. Problem Reformulation

Although the formulation of the LQG problem with memory limitation in the previous subsection clarifies its relationship with that of the LQG problem without memory limitation, it is inconvenient for further mathematical investigations. In order to resolve this problem, we reformulate the LQG problem with memory limitation based on the extended state \( s_t \) (17). The formulation in this subsection is simpler and more general than that in the previous subsection.

In the LQG problem with memory limitation, the extended state SDE (18) is provided as follows:

\[
ds_t = (\tilde{A}(t)s_t + \tilde{B}(t)\tilde{u}_t) dt + \tilde{\sigma}(t) d\tilde{\omega}_t, \tag{51}
\]
where $s_0$ obeys the Gaussian distribution $p_0(s_0) := \mathcal{N}(s_0|\mu_0,\Sigma_0)$, $\omega_t \in \mathbb{R}^{d\omega}$ is the standard Wiener process, and $\tilde{u}_t = \tilde{u}(t, z_t) \in \mathbb{R}^{d\tilde{u}}$ is the control. The extended state SDE (51) includes the previous state, observation, and memory SDEs (30), (31) and (48) as a special case because they can be represented as follows:

$$ds_1 = \left( \begin{array}{c} A \ O \\ \kappa H \ O \end{array} \right) s_1 + \left( \begin{array}{c} B \ O \\ O \ I \end{array} \right) \tilde{u}_1 dt + \left( \begin{array}{c} \sigma \ O \\ O \ \kappa \gamma \ \eta \end{array} \right) \left( \begin{array}{c} d\omega_t \\ d\nu_t \\ d\xi_t \end{array} \right),$$  

(52)

where $p_0(s_0) = p_0(x_0)p_0(z_0)$.

The objective function (21) is provided by the following expected cumulative cost function:

$$J[\tilde{u}] := \mathbb{E}_{p(s_0,t,\tilde{u})} \left[ \int_0^T \left( s_t^\top \tilde{Q}(t)s_t + \tilde{u}_t^\top \tilde{R}(t)\tilde{u}_t \right) dt + s_T^\top \tilde{P}s_T \right],$$  

(53)

where $\tilde{Q}(t) \succeq O$, $\tilde{R}(t) > O$, and $\tilde{P} \succeq O$. This objective function (53) includes the previous objective function (49) as a special case because it can be represented as follows:

$$J[\tilde{u}] = \mathbb{E}_{p(s_0,t,\tilde{u})} \left[ \int_0^T \left( s_t^\top \left( \begin{array}{c} Q \\ O \\ O \end{array} \right) s_t + \tilde{u}_t^\top \left( \begin{array}{c} R \\ O \\ \Pi \end{array} \right) \tilde{u}_t \right) dt + s_T^\top \left( \begin{array}{c} P \\ O \\ O \end{array} \right) s_T \right].$$  

(54)

The objective of the LQG problem with memory limitation is to find the optimal control function $\tilde{u}^\ast$ that minimizes the objective function $J[\tilde{u}]$, as follows:

$$\tilde{u}^\ast := \arg\min_{\tilde{u}} J[\tilde{u}].$$  

(55)

In the following subsection, we mainly consider the formulation of this subsection rather than that of the previous subsection because it is simpler and more general. Moreover, we omit $\tilde{\cdot}$ for notational simplicity.

6.3. Derivation of Optimal Control Function

In this subsection, we derive the optimal control function of the LQG problem with memory limitation by applying Theorem 1. In the LQG problem with memory limitation, the probability of the extended state $s$ at time $t$ is provided by the Gaussian distribution $p_t(s) = \mathcal{N}(s|\mu(t),\Sigma(t))$. By defining the stochastic extended state $\tilde{s} := s - \mu$, $\mathbb{E}_{p_t(\tilde{s}|z)}[\tilde{s}]$ is provided as follows:

$$\mathbb{E}_{p_t(\tilde{s}|z)}[\tilde{s}] = K(t)\tilde{s} + \mu(t),$$  

(56)

where $K(t)$ is defined by

$$K(t) := \left( \begin{array}{cc} O & \Sigma_{xz}(t)\Sigma_{zz}^{-1}(t) \\ O & I \end{array} \right).$$  

(57)

By applying Theorem 1 to the LQG problem with memory limitation, we obtain the following theorem:

**Theorem 3.** In the LQG problem with memory limitation, the optimal control function of ML-POSC is provided by

$$u^\ast(t, z) = -R^{-1}\bar{B}^\top (\Pi K \tilde{s} + \Psi \mu),$$  

(58)
where $K(t)$ (57) depends on $\Sigma(t)$, and $\mu(t)$ and $\Sigma(t)$ are the solutions of the following ordinary differential equations:

\[
\frac{d\mu}{dt} = \left( A - BR^{-1}B^T \Psi \right) \mu, \tag{59}
\]

\[
\frac{d\Sigma}{dt} = \sigma \sigma^T + \left( A - BR^{-1}B^T \Psi \right) \Sigma + \Sigma \left( A - BR^{-1}B^T \Psi \right)^T, \tag{60}
\]

where $\mu(0) = \mu_0$ and $\Sigma(0) = \Sigma_0$, while $\Psi(t)$ and $\Pi(t)$ are the solutions of the following ordinary differential equations:

\[
-\frac{d\Psi}{dt} = Q + A^T \Psi + \Psi A - \Psi BR^{-1}B^T \Psi, \tag{61}
\]

\[
-\frac{d\Pi}{dt} = Q + A^T \Pi + \Pi A - \Pi BR^{-1}B^T \Pi + (I - K)^T \Pi BR^{-1}B^T \Pi (I - K), \tag{62}
\]

where $\Psi(T) = \Pi(T) = P$.

**Proof.** The proof is shown in Appendix E. \(\square\)

Here, (61) is the Riccati equation [9,10,23], which appears in the LQG problem without memory limitation as well (37). In contrast, (62) is a new equation of the LQG problem with memory limitation, which in this paper we call the partially observable Riccati equation. Because estimation and control are not clearly separated in the LQG problem with memory limitation, the Riccati Equation (61) for control is modified to include estimation, which corresponds to the partially observable Riccati Equation (62). As a result, the partially observable Riccati Equation (62) is able to improve estimation as well as control.

In order to support this interpretation, we analyze the partially observable Riccati equation (62) by comparing it with the Riccati Equation (61). Because only the last term of (62) is different from (61), we denote it as follows:

\[
Q := (I - K)^T \Pi BR^{-1}B^T \Pi (I - K). \tag{63}
\]

$Q$ can be calculated as follows:

\[
Q = \begin{pmatrix}
P_{xx} & P_{xz} \\
-\Sigma_{zz}^{-1} \Sigma_{xz} & \Sigma_{zz}^{-1}
\end{pmatrix}, \tag{64}
\]

where $P_{xx} := (\Pi BR^{-1}B^T \Pi)_{xx}$. Because $P_{xx} \succeq O$ and $\Sigma_{zz}^{-1} \Sigma_{xz} \Sigma_{xx} \Sigma_{zz}^{-1} \succeq O$, $P_{xx}$ and $\Pi_{zz}$ may be larger than $\Psi_{xx}$ and $\Psi_{zz}$, respectively. Because $\Pi_{xx}$ and $\Pi_{zz}$ are the negative feedback gains of the state $x$ and the memory $z$, respectively, $Q$ may decrease $\Sigma_{zz}$ and $\Sigma_{zz}$. Moreover, when $\Sigma_{zz}$ is positive/negative, $\Pi_{zz}$ may be smaller/larger than $\Psi_{zz}$, which may increase/decrease $\Sigma_{zz}$. A similar discussion is possible for $\Sigma_{xx}$, $\Pi_{xx}$, and $\Psi_{xx}$, as $\Sigma$, $\Pi$, and $\Psi$ are symmetric matrices. As a result, $Q$ may decrease the following conditional covariance matrix:

\[
\Sigma_{xz} := \Sigma_{xz} - \Sigma_{zz} \Sigma_{xz}^{-1} \Sigma_{xx}, \tag{65}
\]

which corresponds to the estimation error of the state from the memory. Therefore, the partially observable Riccati equation (62) may improve estimation as well as control, which is different from the Riccati Equation (61).

Because the problem in Section 6.1 is specialized more than that in Section 6.2, we can carry out a more specific discussion. In the problem in Section 6.1, $\Psi_{xz}$ is the same as the solution of the Riccati equation of the conventional POSC (37), and $\Psi_{xx} = \Psi_{zz} = O$, and $\Psi_{xz} = O$ are satisfied. As a result, the memory control does not appear in the Riccati equation of ML-POSC (61). In contrast, because of the last term of the partially observable Riccati Equation (62), $\Pi_{xx}$ is not the solution of the Riccati Equation (37), and $\Pi_{xz} \neq O$, etc.
\( \Pi_{zx} \neq O \), and \( \Pi_{zz} \neq O \) are satisfied. As a result, the memory control appears in the partially observable Riccati Equation (62), which may improve the state estimation.

6.4. Comparison with Completely Observable Stochastic Control

In this subsection, we compare ML-POSC with COSC of the extended state. By applying Proposition 2 in the LQG problem, the optimal control function of COSC of the extended state can be obtained as follows:

**Proposition 4 ([10,23]).** In the LQG problem, the optimal control function of COSC of the extended state is provided by

\[
\begin{align*}
u^*(t,s) &= -R^{-1}B^\top \Psi_s = -R^{-1}B^\top (\Psi \hat{s} + \Psi \mu),
\end{align*}
\]

where \( \Psi(t) \) is the solution of the Riccati Equation (61).

**Proof.** The proof is shown in [10,23]. \qed

The optimal control function of COSC of the extended state (66) can be derived intuitively from that of ML-POSC (58). In ML-POSC, \( K\hat{s} = E_{p_t(x|z\hat{s})}\hat{s} \) is the estimator of the stochastic extended state. In COSC of the extended state, because the stochastic extended state is completely observable, its estimator is provided by \( \hat{s} \), which corresponds to \( K = I \). By changing the definition of \( K \) from (57) to \( K = I \), the partially observable Riccati Equation (62) is reduced to the Riccati Equation (61), and the optimal control function of ML-POSC (58) is reduced to that of COSC (66). As a result, the optimal control function of ML-POSC (58) can be interpreted as the generalization of that of COSC (66).

While the second term is the same between (58) and (66), the first term is different. The second term is the control of the expected extended state \( \mu \), which does not depend on the realization. In contrast, the first term is the control of the stochastic extended state \( \hat{s} \), which depends on the realization. The first term has two different points: (i) The estimators of the stochastic extended state in COSC and ML-POSC are provided by \( \hat{s} \) and \( K\hat{s} = E_{p_t(x|z\hat{s})}\hat{s} \), respectively, which is reasonable because ML-POSC needs to estimate the state from the memory; and (ii) The control gains of the stochastic extended state in COSC and ML-POSC are provided by \( \Psi \) and \( \Pi \), respectively. While \( \Psi \) improves only control, \( \Pi \) improves estimation as well as control.

6.5. Numerical Algorithm

In the LQG problem, the partial differential equations are reduced to the ordinary differential equations. The FP Equation (23) is reduced to (59) and (60), and the HJB Equation (28) is reduced to (61) and (62). As a result, the optimal control function (58) can be obtained more easily in the LQG problem.

The Riccati Equation (61) can be solved backwards from the terminal condition. In contrast, the partially observable Riccati Equation (62) cannot be solved in the same way as the Riccati Equation (61), as it depends on the forward equation of \( \Sigma \) (60) through \( K \) (57). Because the forward equation of \( \Sigma \) (60) depends on the backward equation of \( \Pi \) (62) as well, they must be solved simultaneously.

A similar problem appears in the mean-field game and control, and numerous numerical methods have been developed to deal with it [33]. In this paper, we solve the system of (60) and (62) using the forward–backward sweep method, which computes (60) and (62) alternately [33,34]. In ML-POSC, the convergence of the forward–backward sweep method is guaranteed [34].

7. Numerical Experiments

In this section, we demonstrate the effectiveness of ML-POSC using numerical experiments on the LQG problem with memory limitation as well as on the non-LQG problem.
7.1. LQG Problem with Memory Limitation

In this subsection, we show the significance of the partially observable Riccati Equation (62) by a numerical experiment of the LQG problem with memory limitation. We consider the state $x_t \in \mathbb{R}$, the observation $y_t \in \mathbb{R}$, and the memory $z_t \in \mathbb{R}$, which evolve by the following SDEs:

$$
    dx_t = (x_t + u_t)dt + d\omega_t,
$$

$$
    dy_t = x_t dt + dv_t,
$$

$$
    dz_t = v_t dt + dy_t,
$$

where $x_0$ and $z_0$ obey standard Gaussian distributions, $y_0$ is an arbitrary real number, $\omega_t \in \mathbb{R}$ and $v_t \in \mathbb{R}$ are independent standard Wiener processes, and $u_t = u(t, z_t) \in \mathbb{R}$ and $v_t = v(t, z_t) \in \mathbb{R}$ are the controls. The objective function to be minimized is provided as follows:

$$
    J[u, v] := \mathbb{E} \left[ \int_0^{10} (x_t^2 + u_t^2 + v_t^2) dt \right].
$$

Therefore, the objective of this problem is to minimize the state variance by the small state and memory controls. Because this problem includes the memory control cost, it corresponds to the LQG problem with memory limitation.

Figure 2a–c shows the trajectories of $\Psi$ and $\Pi$; $\Pi_{xx}$ and $\Pi_{zz}$ are larger than $\Psi_{xx}$ and $\Psi_{zz}$, respectively, and $\Pi_{xz}$ is smaller than $\Psi_{xz}$, which is consistent with our discussion in Section 6.3. Therefore, the partially observable Riccati equation may reduce the estimation error of the state from the memory. Moreover, while the memory control does not appear in the Riccati equation ($\Psi_{xz} = \Psi_{zz} = 0$), it appears in the partially observable Riccati equation ($\Pi_{xz} \neq 0, \Pi_{zz} \neq 0$), which is consistent with our discussion in Section 6.3. As a result, the memory control plays an important role in estimating the state from the memory.

![Figure 2](image-url)

**Figure 2.** Numerical simulation of the LQG problem with memory limitation. (a–c) Trajectories of the elements of $\Psi(t) \in \mathbb{R}^{2 \times 2}$ and $\Pi(t) \in \mathbb{R}^{2 \times 2}$. Because $\Psi_{xz}(t) = \Psi_{zz}(t)$ and $\Pi_{xx}(t) = \Pi_{zz}(t)$, $\Psi_{xz}(t)$ and $\Pi_{xz}(t)$ are not visualized. (d–f) Stochastic behaviors of the state $x_t$ (d), the memory $z_t$ (e), and the cumulative cost (f) for 100 samples. The expectation of the cumulative cost at $t = 10$ corresponds to the objective function (70). Blue and orange curves are controlled by (71) and (58), respectively.
In order to clarify the significance of the partially observable Riccati Equation (62), we compare the performance of the optimal control function (58) with that of the following control function:

$$u^\Psi(t,z) = -R^{-1}B^T(\Psi\hat{s} + \Psi\mu), \quad (71)$$

in which $\Pi$ is replaced with $\Psi$. This result is shown in Figure 2d–f. In the control function (71), the distributions of the state and the memory are unstable, and the cumulative cost diverges. By contrast, in the optimal control function (58), the distributions of the state and memory are stable, and the cumulative cost is smaller. This result indicates that the partially observable Riccati Equation (62) plays an important role in the LQG problem with memory limitation.

### 7.2. Non-LQG Problem

In this subsection, we investigate the potential effectiveness of ML-POSC for a non-LQG problem by comparing it with the local LQG approximation of the conventional POSC [3,4]. We consider the state $x_t \in \mathbb{R}$ and the observation $y_t \in \mathbb{R}$, which evolve according to the following SDEs:

$$dx_t = u_t dt + d\omega_t, \quad (72)$$
$$dy_t = x_t dt + d\nu_t, \quad (73)$$

where $x_0$ obeys the Gaussian distribution $p_0(x_0) = \mathcal{N}(x_0|0,0.01)$, $y_0$ is an arbitrary real number, $\omega_t \in \mathbb{R}$ and $\nu_t \in \mathbb{R}$ are independent standard Wiener processes, and $u_t = u(t,y_0:t) \in \mathbb{R}$ is the control. The objective function to be minimized is provided as follows:

$$J[u] := \mathbb{E}\left[\int_0^1 \left(Q(t,x) + u_t^2\right) dt + 10x_1^2\right], \quad (74)$$

where

$$Q(t,x) := \begin{cases} 1000 & (0.3 \leq t \leq 0.6, 0.1 \leq |x| \leq 2.0), \\ 0 & \text{(others)}. \end{cases} \quad (75)$$

The cost function is high on the black rectangles in Figure 3a, which represent the obstacles. In addition, the terminal cost function is the lowest on the black cross in Figure 3a, which represents the desirable goal. Therefore, the system should avoid the obstacles and reach the goal with the small control. Because the cost function is non-quadratic, it is a non-LQG problem, which cannot be solved exactly by the conventional POSC.

Figure 3. Numerical simulation of the non-LQG problem for the local LQG approximation (blue) and ML-POSC (orange). (a) Stochastic behaviors of state $x_t$ for 100 samples. The black rectangles and cross represent the obstacles and goal, respectively. (b) The objective function (74), computed from 100 samples.
In the local LQG approximation of the conventional POSC [3,4], the Zakai equation and the Bellman equation are locally approximated by the Kalman filter and the Riccati equation, respectively. Because the Bellman equation is reduced to the Riccati equation, the local LQG approximation can be solved numerically even in the non-LQG problem.

ML-POSC determines the control \( u_t \in \mathbb{R} \) based on the memory \( z_t \in \mathbb{R} \), i.e., \( u_t = u(t, z_t) \). The memory dynamics is formulated with the following SDE:

\[
dz_t = dy_t,
\]

where \( p_0(z_0) = \mathcal{N}(z_0|0,0.01) \). For the sake of simplicity, the memory control is not considered.

Figure 3 is the numerical result comparing the local LQG approximation and ML-POSC. Because the local LQG approximation reduces the Bellman equation to the Riccati equation by ignoring non-LQG information, it cannot avoid the obstacles, which results in a higher objective function. In contrast, because ML-POSC reduces the Bellman equation to the HJB equation while maintaining non-LQG information, it can avoid the obstacles, which results in a lower objective function. Therefore, our numerical experiment shows that ML-POSC can be superior to local LQG approximation.

8. Discussion

In this work, we propose ML-POSC, which is an alternative theoretical framework to the conventional POSC. ML-POSC first formulates the finite-dimensional and stochastic memory dynamics explicitly, then optimizes the memory dynamics considering the memory cost. As a result, unlike the conventional POSC, ML-POSC can consider memory limitation as well as incomplete information. Furthermore, because the optimal control function of ML-POSC is obtained by solving the system of HJB-FP equations, ML-POSC can be solved in practice even in non-LQG problems. ML-POSC can generalize the LQG problem to include memory limitation. Because estimation and control are not clearly separated in the LQG problem with memory limitation, the Riccati equation can be modified to include memory limitation, the Riccati equation can be modified toang to the partially observable Riccati equation, which improves estimation as well as control. Furthermore, ML-POSC can provide a better result than the local LQG approximation in a non-LQG problem, as ML-POSC reduces the Bellman equation while maintaining non-LQG information.

ML-POSC is effective for the state estimation problem as well, which is a part of the POSC problem. Although the state estimation problem can be solved in principle by the Zakai equation [38–40], it cannot be solved directly, as the Zakai equation is infinite-dimensional. In order to resolve this problem, a particle filter is often used to approximate the infinite-dimensional Zakai equation as a finite number of particles [38–40]. However, because the performance of the particle filter is guaranteed only in the limit of a large number of particles, a particle filter may not be practical in cases where the available memory size is severely limited. Furthermore, a particle filter cannot take the memory noise and cost into account. ML-POSC resolves these problems, as it can optimize the state estimation under memory limitation.

ML-POSC may be extended from a single-agent system to a multi-agent system. POSC of a multi-agent system is called decentralized stochastic control (DSC) [41–43], which consists of a system and multiple controllers. In DSC, each controller needs to estimate the controls of the other controllers as well as the state of the system, which is essentially different from the conventional POSC. Because the estimation among the controllers is generally intractable, the conventional POSC approach cannot be straightforwardly extended to DSC. In contrast, ML-POSC compresses the observation history into the finite-dimensional memory, which simplifies estimation among the controllers. Therefore, ML-POSC may provide an effective approach to DSC. Actually, the finite-state controller, the idea of which is similar with ML-POSC, plays a key role in extending POMDP from a single-agent system to a multi-agent system [22,44–48]. ML-POSC may be extended to a multi-agent system in a similar way as a finite-state controller.
ML-POSC can be naturally extended to the mean-field control setting [28–30] because ML-POSC is solved based on the mean-field control theory. Therefore, ML-POSC can be applied to an infinite number of homogeneous agents. Furthermore, ML-POSC can be extended to a risk-sensitive setting, as this is a special case of the mean-field control setting [28–30]. Therefore, ML-POSC can consider the variance of the cost as well as its expectation.

Nonetheless, more efficient algorithms are needed in order to solve ML-POSC with a high-dimensional state and memory. In the mean-field game and control, neural network-based algorithms have recently been proposed which can solve high-dimensional problems efficiently [49,50]. By extending these algorithms, it might be possible to solve high-dimensional ML-POSC efficiently. Furthermore, unlike the mean-field game and control, the coupling of HJB-FP equations is limited to the optimal control function in ML-POSC. By exploiting this property, more efficient algorithms for ML-POSC may be proposed [34].

Author Contributions: Conceptualization, Formal analysis, Funding acquisition, Writing—original draft: T.T. and T.J.K.; Software, Visualization: T.T. All authors have read and agreed to the published version of the manuscript.

Funding: The first author received a JSPS Research Fellowship (Grant No. 21J20436). This work was supported by JSPS KAKENHI (Grant No. 19H05799) and JST CREST (Grant No. JPMJCR2011).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: We thank Kenji Kashima and Kaito Ito for useful discussions.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations
The following abbreviations are used in this manuscript:

- COSC Completely Observable Stochastic Control
- POSC Partially Observable Stochastic Control
- ML-POSC Memory-Limited Partially Observable Stochastic Control
- POMDP Partially Observable Markov Decision Process
- DSC Decentralized Stochastic Control
- LQG Linear-Quadratic-Gaussian
- HJB Hamilton–Jacobi–Bellman
- FP Fokker–Planck
- SDE Stochastic Differential Equation

Appendix A. Proof of Lemma 1

We define the value function \( V(t, p) \) as follows:

\[
V(t, p) := \min_{u_t, p_T} \left[ \int_t^T f(t, p_\tau, u_\tau) d\tau + g_T(p_T) \right],
\]

where \( \{p_\tau | \tau \in [t, T]\} \) is the solution of the FP Equation (23), where \( p_t = p \). Then, \( V(t, p) \) can be calculated as follows:

\[
V(t, p) = \min_u \left[ f(t, p, u) dt + V(t + dt, p + \mathcal{L} p dt) \right]
= \min_u \left[ f(t, p, u) dt + V(t, p) + \frac{\partial V(t, p)}{\partial p} dt + \left( \int f(t, p) \frac{\delta V(t, p)}{\delta p} (s) \mathcal{L} p(s) ds \right) dt \right].
\]
By rearranging the above equation, the following equation is obtained:

$$- \frac{\partial V(t, p)}{\partial t} = \min_u \left[ f(t, p, u) + \int \frac{\partial V(t, p)}{\partial p} (s) \mathcal{L}^s p(s) ds \right]. \quad (A3)$$

Because

$$\int \frac{\partial V(t, p)}{\partial p} (s) \mathcal{L}^s p(s) ds = \int p(s) \mathcal{L} \frac{\partial V(t, p)}{\partial p} (s) ds,$$

the following equation is obtained:

$$- \frac{\partial V(t, p)}{\partial t} = \min_u \int p(s) \left[ f(t, s, u) + \mathcal{L} \frac{\partial V(t, p)}{\partial p} (s) \right] ds. \quad (A5)$$

From the definition of the Hamiltonian $\mathcal{H}$ (10), the following Bellman equation is obtained:

$$- \frac{\partial V(t, p)}{\partial t} = \min_u \mathbb{E}_{p(s)} \left[ \mathcal{H} \left( t, s, u, \frac{\partial V(t, p)}{\partial p} (s) \right) \right]. \quad (A6)$$

Because the control $u$ is the function of the memory $z$ in ML-POSC, the minimization by $u$ can be exchanged with the expectation by $p(z)$ as follows:

$$- \frac{\partial V(t, p)}{\partial t} = \mathbb{E}_{p(z)} \left[ \min_u \mathbb{E}_{p(x|z)} \left[ \mathcal{H} \left( t, s, u, \frac{\partial V(t, p)}{\partial p} (s) \right) \right] \right]. \quad (A7)$$

Because the optimal control function is provided by the right-hand side of the Bellman Equation (A7) [10], the optimal control function is provided by

$$u^*(t, z, p) = \arg\min_u \mathbb{E}_{p(x|z)} \left[ \mathcal{H} \left( t, s, u, \frac{\partial V(t, p)}{\partial p} (s) \right) \right]. \quad (A8)$$

Because the FP Equation (23) is deterministic, the optimal control function is provided by $u^*(t, z) = u^*(t, z, p_t)$.

**Appendix B. Proof of Theorem 1**

We first define

$$W(t, p, s) := \frac{\partial V(t, p)}{\partial p} (s), \quad (A9)$$

which satisfies $W(T, p, s) = g(s)$. Differentiating the Bellman Equation (26) with respect to $p$, the following equation is obtained:

$$- \frac{\partial W(t, p, s)}{\partial t} = \mathcal{H}(t, s, u^*, W) + \mathbb{E}_{p(s')} \left[ \mathcal{L} \frac{\partial W(t, p, s')}{\partial p} (s) \right]. \quad (A10)$$

Because

$$\int p(s') \mathcal{L} \frac{\partial W(t, p, s)}{\partial p} (s') ds' = \int \frac{\partial W(t, p, s)}{\partial p} (s') \mathcal{L}^s p(s') ds', \quad (A11)$$

the following equation is obtained:

$$- \frac{\partial W(t, p, s)}{\partial t} = \mathcal{H}(t, s, u^*, W) + \int \frac{\partial W(t, p, s)}{\partial p} (s') \mathcal{L}^s p(s') ds'. \quad (A12)$$
We then define

\[ w(t, s) := W(t, p_t, s), \]  

(A13)

where \( p_t \) is the solution of the FP Equation (23). The time derivative of \( w(t, s) \) can be calculated as follows:

\[
\frac{\partial w(t, s)}{\partial t} = \frac{\partial W(t, p_t, s)}{\partial t} + \int \frac{\delta W(t, p_t, s)}{\delta p} (s') \frac{\partial p_t(s')}{\partial t} ds'.
\]  

(A14)

By substituting (A12) into (A14), the following equation is obtained:

\[
-\frac{\partial w(t, s)}{\partial t} = \mathcal{H}(t, s, u^*, w) - \int \frac{\delta W(t, p_t, s)}{\delta p} (s') \left( \frac{\partial p_t(s')}{\partial t} - \mathcal{L}_t p_t(s') \right) ds'.
\]  

(A15)

From the FP Equation (23), \( (*) = 0 \) holds. Therefore, the HJB Equation (28) is obtained.

**Appendix C. Proof of Proposition 2**

From the proof of Lemma 1 (Appendix A), the Bellman Equation (A6) is obtained. Because the control \( u \) is the function of the extended state \( s \) in COSC of the extended state, the minimization by \( u \) can be exchanged with the expectation by \( p(s) \) as follows:

\[
-\frac{\partial V(t, p)}{\partial t} = \mathbb{E}_{p(s)} \left[ \min_u \mathcal{H}(t, s, u, \frac{\delta V(t, p)}{\delta p} (s)) \right].
\]  

(A16)

Because the optimal control function is provided by the right-hand side of the Bellman Equation (A16) [10], the optimal control function is provided by

\[
u^*(t, s, p) = \arg\min_u \mathcal{H}(t, s, u, \frac{\delta V(t, p)}{\delta p} (s)).
\]  

(A17)

Because the FP Equation (23) is deterministic, the optimal control function is provided by \( \nu^*(t, s, p) = u^*(t, s, p_t) \). The rest of the proof is the same as the proof of Theorem 1 (Appendix B).

**Appendix D. Proof of Theorem 2**

From Theorem 1, the optimal control functions \( u^*, v^*, \) and \( \kappa^* \) are provided by the minimization of the conditional expectation of the Hamiltonian, as follows:

\[
u^*(t, z), v^*(t, z), \kappa^*(t, z) = \arg\min_{\nu, v, \kappa} \mathbb{E}_{p(\gamma|x)} [\mathcal{H}(t, s, u, v, \kappa, w)].
\]  

(A18)

In the LQG problem of the conventional POSC, the Hamiltonian (10) is provided by

\[
\mathcal{H}(t, s, u, v, \kappa, w) = x^\top Qx + u^\top Ru + \left( \frac{\partial w(t, s)}{\partial x} \right)^\top (Ax + Bu) + \left( \frac{\partial w(t, s)}{\partial z} \right)^\top (v + \kappa Hx)
\]

\[
+ \frac{1}{2} \text{tr} \left\{ \frac{\partial}{\partial x} \left( \frac{\partial w(t, s)}{\partial x} \right)^\top \sigma \sigma^\top \right\}
\]

\[
+ \frac{1}{2} \text{tr} \left\{ \frac{\partial}{\partial z} \left( \frac{\partial w(t, s)}{\partial z} \right)^\top \kappa \gamma \gamma^\top \kappa^\top \right\}.
\]  

(A19)

From

\[
\frac{\partial \mathbb{E}_{p(\gamma|x)} [\mathcal{H}]}{\partial u} = 2Ru + B^\top \mathbb{E}_{p(\gamma|x)} \left[ \frac{\partial w(t, s)}{\partial x} \right],
\]  

(A20)
Entropy 2022, 24, 1599

From the initial condition of the FP equation, calculated as follows: \( \beta \) and \( \Psi \) are satisfied. From the terminal condition of the HJB equation, the optimal control functions are provided by \( v^* \).

We then assume that the following equations are satisfied:

\[ \frac{\partial E_{p(t|z)}[H]}{\partial v} = E_{p(t|z)} \left[ \frac{\partial w(t,s)}{\partial z} \right] \tag{A21} \]
\[ \frac{\partial E_{p(t|z)}[H]}{\partial x} = E_{p(t|z)} \left[ \frac{\partial w(t,s)}{\partial z} x^\top \right] H^\top + \frac{\partial}{\partial z} \left( \frac{\partial w(t,s)}{\partial z} x^\top \right)^\top \kappa \gamma \gamma^\top, \tag{A22} \]

the optimal control functions are provided by

\[ u^*(t,z) = -\frac{1}{2} R^{-1}B^\top E_{p(t|z)} \left[ \frac{\partial w(t,s)}{\partial x} \right], \tag{A23} \]
\[ v^*(t,z) = \begin{cases} +\infty & E_{p(t|z)} \left[ \frac{\partial w(t,s)}{\partial z} \right] < 0, \\ \text{arbitrary} & E_{p(t|z)} \left[ \frac{\partial w(t,s)}{\partial z} \right] = 0, \\ -\infty & E_{p(t|z)} \left[ \frac{\partial w(t,s)}{\partial z} \right] > 0, \end{cases} \tag{A24} \]
\[ \kappa^*(t,z) = -\left( E_{p(t|z)} \left[ \frac{\partial w(t,s)}{\partial z} x^\top \right] \right)^{-1} E_{p(t|z)} \left[ \frac{\partial w(t,s)}{\partial z} x^\top \right] H^\top (\gamma \gamma^\top)^{-1}. \tag{A25} \]

We assume that \( p_t(s) \) is provided by the Gaussian distribution

\[ p_t(s) = \mathcal{N}(s|\mu(t), \Sigma(t)), \tag{A26} \]

and \( w(t,s) \) is provided by the quadratic function

\[ w(t,s) = x^\top \Psi(t)x + (x-z)^\top \Phi(t)(x-z) + \beta(t). \tag{A27} \]

From the initial condition of the FP equation,

\[ \mu(0) = \begin{pmatrix} \mu_x(0) \\ \mu_z(0) \end{pmatrix} = \begin{pmatrix} \mu_{x,0} \\ \mu_{z,0} \end{pmatrix}, \tag{A28} \]
\[ \Sigma(0) = \begin{pmatrix} \Sigma_{xx}(0) & \Sigma_{xz}(0) \\ \Sigma_{xz}(0) & \Sigma_{zz}(0) \end{pmatrix} = \begin{pmatrix} \Sigma_{xx,0} & O \\ O & O \end{pmatrix} \tag{A29} \]

are satisfied. From the terminal condition of the HJB equation, \( \Psi(T) = P, \Phi(T) = O, \) and \( \beta(T) = 0 \) are satisfied. In this case, \( u^*(t,z), E_{p(t|z)}[\partial w(t,s)/\partial z], \) and \( \kappa^*(t,z) \) can be calculated as follows:

\[ u^*(t,z) = -R^{-1}B^\top (\Psi + \Phi) \mu_{x|z} - \Phi z, \tag{A30} \]
\[ E_{p(t|z)} \left[ \frac{\partial w(t,s)}{\partial z} \right] = 2\Phi (z - \mu_{x|z}), \tag{A31} \]
\[ \kappa^*(t,z) = (\Sigma_{x|z} + (\mu_{x|z} - z) \mu_{x|z}^\top) H^\top (\gamma \gamma^\top)^{-1}. \tag{A32} \]

We then assume that the following equations are satisfied:

\[ \mu_x = \mu_z, \tag{A33} \]
\[ \Sigma_{zz} = \Sigma_{xz}. \tag{A34} \]

In this case, \( \mu_{x|z}, \Sigma_{x|z}, u^*(t,z), E_{p(t|z)}[\partial w(t,s)/\partial z], \) and \( \kappa^*(t,z) \) can be calculated as follows:

\[ \mu_{x|z} = z, \tag{A35} \]
\[ \Sigma_{x|z} = \Sigma_{xx} - \Sigma_{zz}, \tag{A36} \]
\[ u^*(t,z) = -R^{-1}B^\top z, \tag{A37} \]
\[ E_{p(t|z)} \left[ \frac{\partial w(t,s)}{\partial z} \right] = 0. \tag{A38} \]
\[ \kappa^*(t, z) = \Sigma_{x,z} H^T (\gamma \gamma^T)^{-1}. \quad (A39) \]

Because \( v^*(t, z) \) is arbitrary when \( E_{p_t(z|z)}[\partial w(t, s)/\partial z] = 0 \), we consider \( v^*(t, z) \) with the following equation:

\[ v^*(t, z) = \left( A - BR^{-1}B^T \Psi - \Sigma_{x|z} H^T (\gamma \gamma^T)^{-1} H \right) z. \quad (A40) \]

In this case, the extended state SDE is provided by the following equation:

\[ ds_t = \bar{A}(t) z_t dt + \bar{\sigma}(t) d\tilde{\omega}_t, \quad (A41) \]

where \( p_0(s) = \mathcal{N}(s|\mu(0), \Sigma(0)) \), and

\[ \bar{A} := \begin{pmatrix} A & -BR^{-1}B^T \Psi \\ \Sigma_{x|z} H^T (\gamma \gamma^T)^{-1} H \\ A - BR^{-1}B^T \Psi - \Sigma_{x|z} H^T (\gamma \gamma^T)^{-1} H \end{pmatrix}, \quad (A42) \]

\[ \bar{\sigma} := \begin{pmatrix} \sigma \\ \Sigma_{x|z} H^T (\gamma \gamma^T)^{-1} \gamma \\ \sigma \end{pmatrix}, \quad \bar{\omega}_t := \begin{pmatrix} \frac{d\omega_t}{dt} \\ \frac{d\omega_t}{dt} \end{pmatrix}. \quad (A43) \]

Because the drift and diffusion coefficients of (A41) are linear and constant with respect to \( s \), respectively, \( p_t(s) \) becomes the Gaussian distribution, which is consistent with our assumptions of (A26), while \( \mu(t) \) and \( \Sigma(t) \) evolve by the following ordinary differential equations:

\[ \frac{d\mu}{dt} = \bar{A} \mu, \quad (A44) \]

\[ \frac{d\Sigma}{dt} = \bar{\sigma} \bar{\sigma}^T + \bar{\Lambda} \Sigma + \Sigma \bar{\Lambda}^T. \quad (A45) \]

If \( \mu_x = \mu_z \) and \( \Sigma_{xz} = \Sigma_{zx} \) are satisfied, \( d\mu_x/dt = d\mu_z/dt \) and \( d\Sigma_{xz}/dt = d\Sigma_{zx}/dt \) are satisfied as well, which is consistent with our assumptions of \( \mu_x = \mu_z \) and \( \Sigma_{xz} = \Sigma_{zx} \).

From \( v^* \) and \( \kappa^* \), the dynamics of \( \mu_{x|z}(t, z_t) = z_t \) is provided by

\[ dz_t = \left( A - BR^{-1}B^T \Psi \right) z_t dt + \Sigma_{x|z} H^T (\gamma \gamma^T)^{-1} (dy_t - Hz_t dt), \quad (A46) \]

where \( z_0 = \mu_{x,0} \). From \( d\Sigma_{xx}/dt \) and \( d\Sigma_{xz}/dt \), the dynamics of \( \Sigma_{x|z} = \Sigma_{xx} - \Sigma_{xz} \) is provided by

\[ \frac{d\Sigma_{x|z}}{dt} = \sigma \sigma^T + A \Sigma_{x|z} + \Sigma_{x|z} A^T - \Sigma_{x|z} H^T (\gamma \gamma^T)^{-1} H \Sigma_{x|z}. \quad (A47) \]

where \( \Sigma_{x|z}(0) = \Sigma_{xx,0} \). We note that (A46) and (A47) correspond to the Kalman filter (35) and (36).

By substituting \( w(t, s), u^*(t, z), v^*(t, z) \), and \( \kappa^*(t, z) \) into the HJB Equation (28), we obtain the following ordinary differential equations:

\[ -\frac{d\Psi}{dt} = Q + A^T \Psi + \Psi A - \Psi BR^{-1}B^T \Psi, \quad (A48) \]

\[ -\frac{d\Phi}{dt} = \left( A - \Sigma_{x|z} H^T (\gamma \gamma^T)^{-1} H \right) \Phi + \Phi \left( A - \Sigma_{x|z} H^T (\gamma \gamma^T)^{-1} H \right)^T + \Psi BR^{-1}B^T \Psi, \quad (A49) \]

\[ -\frac{d\beta}{dt} = \text{tr} \left\{ (\Psi + \Phi) \sigma \sigma^T \right\} + \text{tr} \left\{ \Phi \Sigma_{x|z} H^T (\gamma \gamma^T)^{-1} H \Sigma_{x|z} \right\}, \quad (A50) \]

where \( \Psi(T) = P, \Phi(T) = O, \) and \( \beta(T) = 0 \). If \( \Psi(t), \Phi(t), \) and \( \beta(t) \) satisfy (A48), (A49), and (A50), respectively, the HJB Equation (28) is satisfied, which is consistent with our assumption (A27). We note that (A48) corresponds to the Riccati Equation (37).
Appendix E. Proof of Theorem 3

From Theorem 1, the optimal control function $u^*$ is provided by the minimization of the conditional expectation of the Hamiltonian, as follows:

$$u^*(t,z) = \underset{u}{\text{argmin}} E_{\mu(t|z)}[H(t,s,u,w)].$$  \hfill(A51)

In the LQG problem with memory limitation, the Hamiltonian (10) is provided as follows:

$$H(t,s,u,w) = s^\top Qs + u^\top Ru + \left( \frac{\partial w(t,s)}{\partial s} \right)^\top (As + Bu) + \frac{1}{2} \text{tr} \left\{ \frac{\partial}{\partial s} \left( \frac{\partial w(t,s)}{\partial s} \right)^\top \sigma \sigma^\top \right\}.$$  \hfill(A52)

From

$$\frac{\partial E_{\mu(s|z)}[H(t,s,u,w)]}{\partial u} = 2Ru + B^\top E_{\mu(s|z)} \left[ \frac{\partial w(t,s)}{\partial s} \right],$$  \hfill(A53)

the optimal control function is provided by

$$u^*(t,z) = -\frac{1}{2} R^{-1} B^\top E_{\mu(s|z)} \left[ \frac{\partial w(t,s)}{\partial s} \right].$$  \hfill(A54)

We assume that $p_t(s)$ is provided by the Gaussian distribution

$$p_t(s) = \mathcal{N}(s|\mu(t),\Sigma(t)),$$  \hfill(A55)

and $w(t,s)$ is provided by the quadratic function

$$w(t,s) = s^\top \Pi(t)s + \alpha^\top(t)s + \beta(t).$$  \hfill(A56)

From the initial condition of the FP equation, $\mu(0) = \mu_0$ and $\Sigma(0) = \Sigma_0$ are satisfied. From the terminal condition of the HJB equation, $\Pi(T) = P$, $\alpha(T) = 0$, and $\beta(T) = 0$ are satisfied. In this case, the optimal control function (A54) can be calculated as follows:

$$u^*(t,z) = -\frac{1}{2} R^{-1} B^\top (2\Pi K\dot{s} + 2\Pi \mu + \alpha),$$  \hfill(A57)

where we use (56). Because the optimal control function (A54) is linear with respect to $\dot{s}$, $p_t(s)$ is the Gaussian distribution, which is consistent with our assumption (A55).

By substituting (A56) and (A57) into the HJB Equation (28), we obtain the following ordinary differential equations:

$$-\frac{d\Pi}{dt} = Q + A^\top \Pi + \Pi A - \Pi BR^{-1} B^\top \Pi + Q,$$  \hfill(A58)

$$-\frac{d\alpha}{dt} = (A - BR^{-1} B^\top \Pi)\alpha - 2Q\mu,$$  \hfill(A59)

$$-\frac{d\beta}{dt} = \text{tr} \left\{ \Pi \sigma \sigma^\top \right\} - \frac{1}{4} \alpha^\top BR^{-1} B^\top \alpha + \mu^\top Q\mu.$$  \hfill(A60)

where $Q := (I - K)^\top \Pi BR^{-1} B^\top \Pi (I - K)$. If $\Pi(t)$, $\alpha(t)$, and $\beta(t)$ satisfy (A58), (A59), and (A60), respectively, the HJB Equation (28) is satisfied, which is consistent with our assumption (A56).

By defining $Y(t)$ by $\alpha(t) = 2Y(t)\mu(t)$, the optimal control function (A57) can be calculated as follows:

$$u^*(t,z) = -R^{-1} B^\top (\Pi K\dot{s} + (\Pi + Y)\mu).$$  \hfill(A61)
In this case, $\mu(t)$ obeys the following ordinary differential equation:

$$\frac{d\mu}{dt} = \left(A - BR^{-1}B^\top (\Pi + Y)\right)\mu.$$  \hspace{1cm} (A62)

From $\alpha(t) = 2Y(t)\mu(t)$, (A59) and (A62), $Y(t)$ obeys the following ordinary differential equation:

$$-\frac{dY}{dt} = \left(A - BR^{-1}B^\top \Pi\right)^\top Y + Y\left(A - BR^{-1}B^\top \Pi\right) - YBR^{-1}B^\top Y - Q,$$  \hspace{1cm} (A63)

where $Y(T) = O$.

By defining $\Psi(t) := \Pi(t) + Y(t)$, the optimal control function (A61) can be calculated as follows:

$$u^*(t,z) = -R^{-1}B^\top (\Pi Kz + \Psi \mu).$$  \hspace{1cm} (A64)

From $\Psi(t) = \Pi(t) + Y(t)$, (A58), and (A63), $\Psi(t)$ obeys the following ordinary differential equation:

$$-\frac{d\Psi}{dt} = Q + A^\top \Psi + \Psi A - \Psi BR^{-1}B^\top \Psi,$$  \hspace{1cm} (A65)

where $\Psi(T) = O$. Therefore, the optimal control function (58) is obtained.

References
1. Fox, R.; Tishby, N. Minimum-information LQG control part I: Memoryless controllers. In Proceedings of the 2016 IEEE 55th Conference on Decision and Control (CDC), Las Vegas, NV, USA, 12–14 December 2016; pp. 5610–5616. [CrossRef]
2. Fox, R.; Tishby, N. Minimum-information LQG control Part II: Retentive controllers. In Proceedings of the 2016 IEEE 55th Conference on Decision and Control (CDC), Las Vegas, NV, USA, 12–14 December 2016; pp. 5603–5609. [CrossRef]
3. Li, W.; Todorov, E. An Iterative Optimal Control and Estimation Design for Nonlinear Stochastic System. In Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, Madison, WI, USA, 24–26 July 1998; pp. 211–219.
45. Bernstein, D.S. Bounded Policy Iteration for Decentralized POMDPs. In Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, Edinburgh, UK, 30 July–5 August 2005; pp. 1287–1292.

46. Bernstein, D.S.; Amato, C.; Hansen, E.A.; Zilberstein, S. Policy Iteration for Decentralized Control of Markov Decision Processes. *J. Artif. Intell. Res.* **2009**, *34*, 89–132. [CrossRef]

47. Amato, C.; Bernstein, D.S.; Zilberstein, S. Optimizing Memory-Bounded Controllers for Decentralized POMDPs. In Proceedings of the Twenty-Third Conference on Uncertainty in Artificial Intelligence, Vancouver, BC, Canada, 19–22 July 2007; pp. 1–8.

48. Tottori, T.; Kobayashi, T.J. Forward and Backward Bellman Equations Improve the Efficiency of the EM Algorithm for DEC-POMDP. *Entropy* **2021**, *23*, 551. [CrossRef]

49. Ruthotto, L.; Osher, S.J.; Li, W.; Nurbekyan, L.; Fung, S.W. A machine learning framework for solving high-dimensional mean field game and mean field control problems. *Proc. Natl. Acad. Sci. USA* **2020**, *117*, 9183–9193. [CrossRef]

50. Lin, A.T.; Fung, S.W.; Li, W.; Nurbekyan, L.; Osher, S.J. Alternating the population and control neural networks to solve high-dimensional stochastic mean-field games. *Proc. Natl. Acad. Sci. USA* **2021**, *118*, e2024713118. [CrossRef] [PubMed]