Ensemble Kalman Filter for Reinforcement Learning

SIAM Conference on Uncertainty Quantification (UQ22)

Anant A. Joshi[†], Amirhossein Taghvaei[*], Prashant G. Mehta[†]

[†]University of Illinois at Urbana-Champaign
[*]University of Washington, Seattle

April 14

Outline

Outline

Problem Formulation

- The linear quadratic optimal control problem (LQR):

$$\begin{aligned} \text{minimize} \quad & \int_0^T \left(\frac{1}{2} \|CX_t\|^2 + \frac{1}{2} U_t^\top R U_t \right) \mathrm{d}t + \frac{1}{2} X_T^\top P_f X_T \\ \text{subject to} \quad & \dot{X}_t = AX_t + BU_t =: f(X_t, U_t) \end{aligned}$$

- **Solution:** $U_t = -R^{-1} B^\top P_t X_t$ where P_t solves

(backward Ricatti eq.) $\quad \dot{P}_t = -A^\top P_t - P_t A - C^\top C + P_t B R^{-1} B^\top P_t, \quad P_T = P_f$

- **Two issues with solving Ricatti directly:**
  (1) model parameters may not be known
  (2) computationally $\mathcal{O}(d^2)$ in the dimension d of the state-space

- **Objective:** design a controlled interacting particle systems

$$\dot{X}_t^i = f(X_t^i, u_t^i) + v_t^i, \quad \text{for} \quad i = 1, \ldots, N$$

to approximately solve LQR using only a simulator $(x, u) \mapsto f(x, u)$

Problem Formulation

- The linear quadratic optimal control problem (LQR):

$$\begin{aligned} \text{minimize} \quad & \int_0^T \left(\frac{1}{2} \|CX_t\|^2 + \frac{1}{2} U_t^\top R U_t \right) \mathrm{d}t + \frac{1}{2} X_T^\top P_f X_T \\ \text{subject to} \quad & \dot{X}_t = AX_t + BU_t =: f(X_t, U_t) \end{aligned}$$

- **Solution:** $U_t = -R^{-1} B^\top P_t X_t$ where P_t solves

(backward Ricatti eq.) $\quad \dot{P}_t = -A^\top P_t - P_t A - C^\top C + P_t B R^{-1} B^\top P_t, \quad P_T = P_f$

- **Two issues with solving Ricatti directly:**
  (1) model parameters may not be known
  (2) computationally $\mathcal{O}(d^2)$ in the dimension d of the state-space
- **Objective:** design a controlled interacting particle systems

$$\dot{X}_t^i = f(X_t^i, u_t^i) + v_t^i, \quad \text{for} \quad i = 1, \ldots, N$$

to approximately solve LQR using only a simulator $(x, u) \mapsto f(x, u)$

Problem Formulation

- The linear quadratic optimal control problem (LQR):

$$\begin{aligned} \text{minimize} \quad & \int_0^T \left(\frac{1}{2} \|CX_t\|^2 + \frac{1}{2} U_t^\top R U_t \right) \mathrm{d}t + \frac{1}{2} X_T^\top P_f X_T \\ \text{subject to} \quad & \dot{X}_t = AX_t + BU_t =: f(X_t, U_t) \end{aligned}$$

- **Solution:** $U_t = -R^{-1} B^\top P_t X_t$ where P_t solves

(backward Ricatti eq.) $\quad \dot{P}_t = -A^\top P_t - P_t A - C^\top C + P_t B R^{-1} B^\top P_t, \quad P_T = P_f$

- **Two issues with solving Ricatti directly:**
  (1) model parameters may not be known
  (2) computationally $\mathcal{O}(d^2)$ in the dimension d of the state-space
- **Objective:** design a controlled interacting particle systems

$$\dot{X}_t^i = f(X_t^i, u_t^i) + v_t^i, \quad \text{for} \quad i = 1, \ldots, N$$

to approximately solve LQR using only a simulator $(x, u) \mapsto f(x, u)$

Problem Formulation

- The linear quadratic optimal control problem (LQR):

$$\text{minimize} \quad \int_0^T \left(\frac{1}{2}\|CX_t\|^2 + \frac{1}{2}U_t^\top R U_t \right) \mathrm{d}t + \frac{1}{2} X_T^\top P_f X_T$$
$$\text{subject to} \quad \dot{X}_t = AX_t + BU_t =: f(X_t, U_t)$$

- **Solution:** $U_t = -R^{-1}B^\top P_t X_t$ where P_t solves

(backward Ricatti eq.) $\dot{P}_t = -A^\top P_t - P_t A - C^\top C + P_t B R^{-1} B^\top P_t, \quad P_T = P_f$

- **Two issues with solving Ricatti directly:**
  1. (1) model parameters may not be known
  2. (2) computationally $\mathcal{O}(d^2)$ in the dimension d of the state-space
- **Objective:** design a controlled interacting particle systems

$$\dot{X}_t^i = f(X_t^i, u_t^i) + v_t^i, \quad \text{for} \quad i = 1, \ldots, N$$

to approximately solve LQR using only a simulator $(x, u) \mapsto f(x, u)$

Related works

Reinforcement learning (RL) approaches:

- Coarse-ID Control (Tu & Recht 2019; Dean et. al. 2020)

  System ID + robust optimization

- Policy gradient method (Fazel el al. 2018; Mohammadi et al. 2021)

  (1) Assumes $u = Kx$ and iteratively learns K using gradient descent

  (2) Each iteration involves simulating

$$X_t^i = f(X_t^i, (K + \delta K^i) X_t^i), \quad \text{for} \quad i = 1, \ldots, N$$

  to approximate the gradient of the cost with respect to K

  (3) requires initialization with stabilizing gain

Related works

Reinforcement learning (RL) approaches:

- Coarse-ID Control (Tu & Recht 2019; Dean et. al. 2020)

  System ID + robust optimization

- Policy gradient method (Fazel el al. 2018; Mohammadi et al. 2021)

  (1) Assumes $u = Kx$ and iteratively learns K using gradient descent

  (2) Each iteration involves simulating

  $$X_t^i = f(X_t^i, (K + \delta K^i)X_t^i), \quad \text{for} \quad i = 1, \ldots, N$$

  to approximate the gradient of the cost with respect to K

  (3) requires initialization with stabilizing gain

Related works

Reinforcement learning (RL) approaches:

- Coarse-ID Control (Tu & Recht 2019; Dean et. al. 2020)

  System ID + robust optimization

- Policy gradient method (Fazel el al. 2018; Mohammadi et al. 2021)

  (1) Assumes $u = Kx$ and iteratively learns K using gradient descent

  (2) Each iteration involves simulating

  $$X_t^i = f(X_t^i, (K + \delta K^i)X_t^i), \quad \text{for} \quad i = 1, \ldots, N$$

  to approximate the gradient of the cost with respect to K

  (3) requires initialization with stabilizing gain

Related works

Reinforcement learning (RL) approaches:

- Coarse-ID Control (Tu & Recht 2019; Dean et. al. 2020)

  System ID + robust optimization

- Policy gradient method (Fazel el al. 2018; Mohammadi et al. 2021)

  (1) Assumes $u = Kx$ and iteratively learns K using gradient descent

  (2) Each iteration involves simulating

$$X_t^i = f(X_t^i, (K + \delta K^i)X_t^i), \quad \text{for} \quad i = 1, \ldots, N$$

  to approximate the gradient of the cost with respect to K

  (3) requires initialization with stabilizing gain

Related works

Reinforcement learning (RL) approaches:

- Coarse-ID Control (Tu & Recht 2019; Dean et. al. 2020)

  System ID + robust optimization

- Policy gradient method (Fazel el al. 2018; Mohammadi et al. 2021)

  (1) Assumes $u = Kx$ and iteratively learns K using gradient descent

  (2) Each iteration involves simulating

  $$X_t^i = f(X_t^i, (K + \delta K^i)X_t^i), \quad \text{for} \quad i = 1, \ldots, N$$

  to approximate the gradient of the cost with respect to K

  (3) requires initialization with stabilizing gain

Our approach: EnKF type algorithm

- Simulate a controlled system of particles

$$\mathrm{d}X_t^i = \underbrace{f(X_t^i, \frac{\mathrm{d}\overleftarrow{\eta}_t^i}{\mathrm{d}t})\,\mathrm{d}t}_{\text{dynamics simulator}} + \underbrace{\frac{1}{2}\Sigma_t^{(N)} C^\top C X_t^i\,\mathrm{d}t}_{\text{RL correction}}, \quad X_T^i \overset{\text{i.i.d}}{\sim} \mathcal{N}(0, P_T^{-1})$$

where $\Sigma_t^{(N)} = \frac{1}{N}\sum_{i=1}^N X_t^i (X_t^i)^\top$ and $\overleftarrow{\eta}_t^i$ is a backward Wiener process

- Solution to the Ricatti eq. is approximated using empirical covariance $\Sigma_t^{(N)}$
- Computationally more efficient in comparison to related works

Our approach: EnKF type algorithm

- Simulate a controlled system of particles

$$\mathrm{d}X_t^i = \underbrace{f(X_t^i, \frac{\mathrm{d}\overleftarrow{\eta}_t^i}{\mathrm{d}t})\,\mathrm{d}t}_{\text{dynamics simulator}} + \underbrace{\frac{1}{2}\Sigma_t^{(N)} C^\top C X_t^i\,\mathrm{d}t}_{\text{RL correction}}, \quad X_T^i \overset{\text{i.i.d}}{\sim} \mathcal{N}(0, P_T^{-1})$$

where $\Sigma_t^{(N)} = \frac{1}{N}\sum_{i=1}^{N} X_t^i (X_t^i)^\top$ and $\overleftarrow{\eta}_t^i$ is a backward Wiener process

- Solution to the Ricatti eq. is approximated using empirical covariance $\Sigma_t^{(N)}$
- Computationally more efficient in comparison to related works

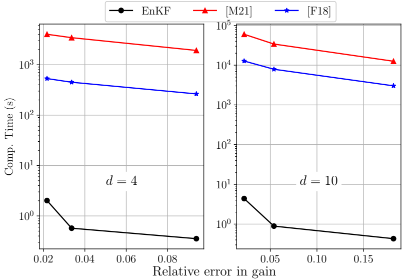
Our approach: EnKF type algorithm

- Simulate a controlled system of particles

$$\mathrm{d}X_t^i = \underbrace{f(X_t^i, \frac{\mathrm{d}\overleftarrow{\eta}_t^i}{\mathrm{d}t})\,\mathrm{d}t}_{\text{dynamics simulator}} + \underbrace{\frac{1}{2}\Sigma_t^{(N)} C^\top C X_t^i\,\mathrm{d}t}_{\text{RL correction}}, \quad X_T^i \overset{\text{i.i.d}}{\sim} \mathcal{N}(0, P_T^{-1})$$

where $\Sigma_t^{(N)} = \frac{1}{N}\sum_{i=1}^N X_t^i (X_t^i)^\top$ and $\overleftarrow{\eta}_t^i$ is a backward Wiener process

- Solution to the Ricatti eq. is approximated using empirical covariance $\Sigma_t^{(N)}$
- Computationally more efficient in comparison to related works



Outline

1. Introduction
2. EnKF for Control
3. Numerical Results
4. Extension to nonlinear setting

Review of EnKF for filtering

- **Linear Gaussian filtering problem:** find the conditional dist. $P(X_t|\mathcal{Z}_t)$

$$\mathrm{d}X_t = AX_t\,\mathrm{d}t + Q^{\frac{1}{2}}\,\mathrm{d}\xi_t =: f(X_t,\xi_t)$$
$$\mathrm{d}Z_t = HX_t\,\mathrm{d}t + \mathcal{R}^{\frac{1}{2}}\,\mathrm{d}W_t$$

- **Kalman-Bucy filter:** conditional dist. is Gaussian $N(m_t,\Sigma_t)$

$$\mathrm{d}m_t = Am_t\,\mathrm{d}t + \Sigma_t H^\top \mathcal{R}^{-1}(\,\mathrm{d}Z_t - Hm_t\,\mathrm{d}t),\quad m_0 = \mathsf{E}(X_0)$$
$$\dot{\Sigma}_t = A\Sigma_t + \Sigma_t A^\top + Q - \Sigma_t H^\top \mathcal{R}^{-1} H\Sigma_t,\quad \Sigma_0 = \mathsf{var}(X_0)$$

- **EnKF algorithm:**
  - Construct a stochastic process $\bar{X}_t$ such that

$$\text{Law}(\bar{X}_t) = N(m_t,\Sigma_t)$$

  - Realize $\bar{X}_t$ with a system of interacting particles $\{X_t^1,\ldots,X_t^N\}$ s.t.

$$\frac{1}{N}\sum_{i=1}^{N}\delta_{X_t^i} \approx \text{Law}(\bar{X}_t)$$

Review of EnKF for filtering

- **Linear Gaussian filtering problem:** find the conditional dist. $P(X_t|\mathcal{Z}_t)$

$$\mathrm{d}X_t = AX_t\,\mathrm{d}t + Q^{\frac{1}{2}}\,\mathrm{d}\xi_t =: f(X_t,\xi_t)$$
$$\mathrm{d}Z_t = HX_t\,\mathrm{d}t + \mathcal{R}^{\frac{1}{2}}\,\mathrm{d}W_t$$

- **Kalman-Bucy filter:** conditional dist. is Gaussian $N(m_t,\Sigma_t)$

$$\mathrm{d}m_t = Am_t\,\mathrm{d}t + \Sigma_t H^\top \mathcal{R}^{-1}(\,\mathrm{d}Z_t - Hm_t\,\mathrm{d}t), \quad m_0 = \mathsf{E}(X_0)$$
$$\dot{\Sigma}_t = A\Sigma_t + \Sigma_t A^\top + Q - \Sigma_t H^\top \mathcal{R}^{-1} H\Sigma_t, \quad \Sigma_0 = \mathsf{var}(X_0)$$

- **EnKF algorithm:**
  - Construct a stochastic process $\bar{X}_t$ such that

$$\mathrm{Law}(\bar{X}_t) = N(m_t,\Sigma_t)$$

  - Realize $\bar{X}_t$ with a system of interacting particles $\{X_t^1,\ldots,X_t^N\}$ s.t.

$$\frac{1}{N}\sum_{i=1}^N \delta_{X_t^i} \approx \mathrm{Law}(\bar{X}_t)$$

Review of EnKF for filtering

- **Linear Gaussian filtering problem:** find the conditional dist. $P(X_t|\mathcal{Z}_t)$

$$\mathrm{d}X_t = AX_t\,\mathrm{d}t + Q^{\frac{1}{2}}\,\mathrm{d}\xi_t =: f(X_t,\xi_t)$$
$$\mathrm{d}Z_t = HX_t\,\mathrm{d}t + \mathcal{R}^{\frac{1}{2}}\,\mathrm{d}W_t$$

- **Kalman-Bucy filter:** conditional dist. is Gaussian $N(m_t,\Sigma_t)$

$$\mathrm{d}m_t = Am_t\,\mathrm{d}t + \Sigma_t H^\top \mathcal{R}^{-1}(\,\mathrm{d}Z_t - Hm_t\,\mathrm{d}t), \quad m_0 = \mathsf{E}(X_0)$$
$$\dot{\Sigma}_t = A\Sigma_t + \Sigma_t A^\top + Q - \Sigma_t H^\top \mathcal{R}^{-1} H\Sigma_t, \quad \Sigma_0 = \mathsf{var}(X_0)$$

- **EnKF algorithm:**
  - Construct a stochastic process $\bar{X}_t$ such that

$$\text{Law}(\bar{X}_t) = N(m_t,\Sigma_t)$$

  - Realize $\bar{X}_t$ with a system of interacting particles $\{X_t^1,\ldots,X_t^N\}$ s.t.

$$\frac{1}{N}\sum_{i=1}^N \delta_{X_t^i} \approx \text{Law}(\bar{X}_t)$$

Review of EnKF: mean-field process and particle system

- **Mean-field process:**

$$\mathrm{d}\bar{X}_t = \underbrace{f(\bar{X}_t, \bar{\xi}_t)}_{\text{dynamics}} + \underbrace{\bar{\Sigma}_t H^\top \mathcal{R}^{-1}\big(\,\mathrm{d}Z_t - \frac{H\bar{X}_t + H\bar{m}_t}{2}\,\mathrm{d}t\big)}_{\text{feedback correction}}, \quad \bar{X}_0 \sim \mathcal{N}(m_0, \Sigma_0)$$

where $\bar{m}_t := \mathsf{E}[\bar{X}_t|\mathcal{Z}_t]$ and $\bar{\Sigma}_t := \mathsf{var}(\bar{X}_t|\mathcal{Z}_t)$

- **Exactness:** $\text{Law}(\bar{X}_t) = N(m_t, \Sigma_t)$ given by the Kalman filter. In particular,

$$\bar{m}_t = m_t, \quad \bar{\Sigma}_t = \Sigma_t$$

- **Particle system:**

$$\mathrm{d}X_t^i = f(X_t^i, \xi_t^i) + \Sigma_t^{(N)} H^\top \mathcal{R}^{-1}\big(\,\mathrm{d}Z_t - \frac{HX_t^i + Hm_t^{(N)}}{2}\,\mathrm{d}t\big), \quad X_0^i \sim \mathcal{N}(m_0, \Sigma_0)$$

where $m_t^{(N)}$ and $\Sigma_t^{(N)}$ are empirical approximations of $\bar{m}_t$ and $\bar{\Sigma}_t$.

- **Approximation error:** Under suitable assumptions, we have

$$\mathsf{E}[\|m_t^{(N)} - m_t\|^2] \leq \frac{\text{(const.)}}{N}, \quad \mathsf{E}[\|\Sigma_t^{(N)} - \Sigma_t\|^2] \leq \frac{\text{(const.)}}{N}$$

Review of EnKF: mean-field process and particle system

- **Mean-field process:**

$$\mathrm{d}\bar{X}_t = \underbrace{f(\bar{X}_t, \bar{\xi}_t)}_{\text{dynamics}} + \underbrace{\bar{\Sigma}_t H^\top \mathcal{R}^{-1}\big(\mathrm{d}Z_t - \frac{H\bar{X}_t + H\bar{m}_t}{2}\,\mathrm{d}t\big)}_{\text{feedback correction}}, \quad \bar{X}_0 \sim \mathcal{N}(m_0, \Sigma_0)$$

where $\bar{m}_t := \mathsf{E}[\bar{X}_t|\mathcal{Z}_t]$ and $\bar{\Sigma}_t := \mathsf{var}(\bar{X}_t|\mathcal{Z}_t)$

- **Exactness:** Law$(\bar{X}_t) = N(m_t, \Sigma_t)$ given by the Kalman filter. In particular,

$$\bar{m}_t = m_t, \quad \bar{\Sigma}_t = \Sigma_t$$

- **Particle system:**

$$\mathrm{d}X_t^i = f(X_t^i, \xi_t^i) + \Sigma_t^{(N)} H^\top \mathcal{R}^{-1}\big(\mathrm{d}Z_t - \frac{HX_t^i + Hm_t^{(N)}}{2}\,\mathrm{d}t\big), \quad X_0^i \sim \mathcal{N}(m_0, \Sigma_0)$$

where $m_t^{(N)}$ and $\Sigma_t^{(N)}$ are empirical approximations of $\bar{m}_t$ and $\bar{\Sigma}_t$.

- **Approximation error:** Under suitable assumptions, we have

$$\mathsf{E}[\|m_t^{(N)} - m_t\|^2] \le \frac{\text{(const.)}}{N}, \quad \mathsf{E}[\|\Sigma_t^{(N)} - \Sigma_t\|^2] \le \frac{\text{(const.)}}{N}$$

Review of EnKF: mean-field process and particle system

- **Mean-field process:**

$$\mathrm{d}\bar{X}_t = \underbrace{f(\bar{X}_t, \bar{\xi}_t)}_{\text{dynamics}} + \underbrace{\bar{\Sigma}_t H^\top \mathcal{R}^{-1}(\mathrm{d}Z_t - \frac{H\bar{X}_t + H\bar{m}_t}{2}\mathrm{d}t)}_{\text{feedback correction}}, \quad \bar{X}_0 \sim \mathcal{N}(m_0, \Sigma_0)$$

where $\bar{m}_t := \mathsf{E}[\bar{X}_t | \mathcal{Z}_t]$ and $\bar{\Sigma}_t := \mathsf{var}(\bar{X}_t | \mathcal{Z}_t)$

- **Exactness:** $\text{Law}(\bar{X}_t) = N(m_t, \Sigma_t)$ given by the Kalman filter. In particular,

$$\bar{m}_t = m_t, \quad \bar{\Sigma}_t = \Sigma_t$$

- **Particle system:**

$$\mathrm{d}X_t^i = f(X_t^i, \xi_t^i) + \Sigma_t^{(N)} H^\top \mathcal{R}^{-1}(\mathrm{d}Z_t - \frac{HX_t^i + Hm_t^{(N)}}{2}\mathrm{d}t), \quad X_0^i \sim \mathcal{N}(m_0, \Sigma_0)$$

where $m_t^{(N)}$ and $\Sigma_t^{(N)}$ are empirical approximations of $\bar{m}_t$ and $\bar{\Sigma}_t$.

- **Approximation error:** Under suitable assumptions, we have

$$\mathsf{E}[\|m_t^{(N)} - m_t\|^2] \leq \frac{(\text{const.})}{N}, \quad \mathsf{E}[\|\Sigma_t^{(N)} - \Sigma_t\|^2] \leq \frac{(\text{const.})}{N}$$

Review of EnKF: mean-field process and particle system

- **Mean-field process:**

$$\mathrm{d}\bar{X}_t = \underbrace{f(\bar{X}_t, \bar{\xi}_t)}_{\text{dynamics}} + \underbrace{\bar{\Sigma}_t H^\top \mathcal{R}^{-1}(\,\mathrm{d}Z_t - \frac{H\bar{X}_t + H\bar{m}_t}{2}\,\mathrm{d}t)}_{\text{feedback correction}}, \quad \bar{X}_0 \sim \mathcal{N}(m_0, \Sigma_0)$$

where $\bar{m}_t := \mathsf{E}[\bar{X}_t|\mathcal{Z}_t]$ and $\bar{\Sigma}_t := \mathsf{var}(\bar{X}_t|\mathcal{Z}_t)$

- **Exactness:** $\mathrm{Law}(\bar{X}_t) = N(m_t, \Sigma_t)$ given by the Kalman filter. In particular,

$$\bar{m}_t = m_t, \quad \bar{\Sigma}_t = \Sigma_t$$

- **Particle system:**

$$\mathrm{d}X_t^i = f(X_t^i, \xi_t^i) + \Sigma_t^{(N)} H^\top \mathcal{R}^{-1}(\,\mathrm{d}Z_t - \frac{HX_t^i + Hm_t^{(N)}}{2}\,\mathrm{d}t), \quad X_0^i \sim \mathcal{N}(m_0, \Sigma_0)$$

where $m_t^{(N)}$ and $\Sigma_t^{(N)}$ are empirical approximations of $\bar{m}_t$ and $\bar{\Sigma}_t$.

- **Approximation error:** Under suitable assumptions, we have

$$\mathsf{E}[\|m_t^{(N)} - m_t\|^2] \le \frac{(\text{const.})}{N}, \quad \mathsf{E}[\|\Sigma_t^{(N)} - \Sigma_t\|^2] \le \frac{(\text{const.})}{N}$$

EnKF for Control: Attempt I

- EnKF is a simulation-based algorithm to solve the Ricatti eq. in filtering
- Can we extend EnKF to simulate the Ricatti eq. in LQR:

$$\dot{P}_t = -A^\top P_t - P_t A - C^\top C + P_t B R^{-1} B^\top P_t, \quad P_T = P_f$$

- **Attempt I:** Construct a stochastic process with variance equal to P_t

$$\mathrm{d}\bar{X}_t = -A^\top \bar{X}_t \,\mathrm{d}t + C^\top \,\mathrm{d}\overleftarrow{\eta}_t + \frac{1}{2}\bar{\Sigma}_t B R^{-1} B^\top \bar{X}_t, \quad \bar{X}_T \sim N(0, P_f)$$

  where $\bar{\Sigma}_t = \mathsf{E}[\bar{X}_t \bar{X}_t^\top]$ and $\overleftarrow{\eta}$ is a backward Wiener process
- Using the Itö rule for backward process:

$$\frac{\mathrm{d}}{\mathrm{d}t}\bar{\Sigma}_t = -A^\top \bar{\Sigma}_t - \bar{\Sigma}_t A - C^\top C + \bar{\Sigma}_t B R^{-1} B^\top \bar{\Sigma}_t, \quad \Sigma_T = P_f$$

  Identical to Ricatti eq for P_t, therefore $\bar{\Sigma}_t = P_t$
- However, it can not be implemented using a simulator since $A^\top$ is not available.

EnKF for Control: Attempt I

- EnKF is a simulation-based algorithm to solve the Ricatti eq. in filtering
- Can we extend EnKF to simulate the Ricatti eq. in LQR:

$$\dot{P}_t = -A^\top P_t - P_t A - C^\top C + P_t B R^{-1} B^\top P_t, \quad P_T = P_f$$

- **Attempt I:** Construct a stochastic process with variance equal to P_t

$$\mathrm{d}\bar{X}_t = -A^\top \bar{X}_t \,\mathrm{d}t + C^\top \,\mathrm{d}\overleftarrow{\eta}_t + \frac{1}{2}\bar{\Sigma}_t B R^{-1} B^\top \bar{X}_t, \quad \bar{X}_T \sim N(0, P_f)$$

  where $\bar{\Sigma}_t = \mathsf{E}[\bar{X}_t \bar{X}_t^\top]$ and $\overleftarrow{\eta}$ is a backward Wiener process
- Using the Itö rule for backward process:

$$\frac{\mathrm{d}}{\mathrm{d}t}\bar{\Sigma}_t = -A^\top \bar{\Sigma}_t - \bar{\Sigma}_t A - C^\top C + \bar{\Sigma}_t B R^{-1} B^\top \bar{\Sigma}_t, \quad \Sigma_T = P_f$$

  Identical to Ricatti eq for P_t, therefore $\bar{\Sigma}_t = P_t$
- However, it can not be implemented using a simulator since $A^\top$ is not available.

EnKF for Control: Attempt I

- EnKF is a simulation-based algorithm to solve the Ricatti eq. in filtering
- Can we extend EnKF to simulate the Ricatti eq. in LQR:

$$\dot{P}_t = -A^\top P_t - P_t A - C^\top C + P_t B R^{-1} B^\top P_t, \quad P_T = P_f$$

- **Attempt I:** Construct a stochastic process with variance equal to P_t

$$\mathrm{d}\bar{X}_t = -A^\top \bar{X}_t \,\mathrm{d}t + C^\top \,\mathrm{d}\overleftarrow{\eta}_t + \frac{1}{2}\bar{\Sigma}_t B R^{-1} B^\top \bar{X}_t, \quad \bar{X}_T \sim N(0, P_f)$$

  where $\bar{\Sigma}_t = \mathsf{E}[\bar{X}_t \bar{X}_t^\top]$ and $\overleftarrow{\eta}$ is a backward Wiener process
- Using the Itö rule for backward process:

$$\frac{\mathrm{d}}{\mathrm{d}t}\bar{\Sigma}_t = -A^\top \bar{\Sigma}_t - \bar{\Sigma}_t A - C^\top C + \bar{\Sigma}_t B R^{-1} B^\top \bar{\Sigma}_t, \quad \Sigma_T = P_f$$

  Identical to Ricatti eq for P_t, therefore $\bar{\Sigma}_t = P_t$
- However, it can not be implemented using a simulator since $A^\top$ is not available.

EnKF for Control: Attempt I

- EnKF is a simulation-based algorithm to solve the Ricatti eq. in filtering
- Can we extend EnKF to simulate the Ricatti eq. in LQR:

$$\dot{P}_t = -A^\top P_t - P_t A - C^\top C + P_t B R^{-1} B^\top P_t, \quad P_T = P_f$$

- **Attempt I:** Construct a stochastic process with variance equal to P_t

$$\mathrm{d}\bar{X}_t = -A^\top \bar{X}_t \,\mathrm{d}t + C^\top \,\mathrm{d}\overleftarrow{\eta}_t + \frac{1}{2}\bar{\Sigma}_t B R^{-1} B^\top \bar{X}_t, \quad \bar{X}_T \sim N(0, P_f)$$

  where $\bar{\Sigma}_t = \mathsf{E}[\bar{X}_t \bar{X}_t^\top]$ and $\overleftarrow{\eta}$ is a backward Wiener process
- Using the Itö rule for backward process:

$$\frac{\mathrm{d}}{\mathrm{d}t}\bar{\Sigma}_t = -A^\top \bar{\Sigma}_t - \bar{\Sigma}_t A - C^\top C + \bar{\Sigma}_t B R^{-1} B^\top \bar{\Sigma}_t, \quad \Sigma_T = P_f$$

  Identical to Ricatti eq for P_t, therefore $\bar{\Sigma}_t = P_t$
- However, it can not be implemented using a simulator since $A^\top$ is not available.

EnKF for Control: Attempt I

- EnKF is a simulation-based algorithm to solve the Ricatti eq. in filtering
- Can we extend EnKF to simulate the Ricatti eq. in LQR:

$$\dot{P}_t = -A^\top P_t - P_t A - C^\top C + P_t B R^{-1} B^\top P_t, \quad P_T = P_f$$

- **Attempt I:** Construct a stochastic process with variance equal to P_t

$$\mathrm{d}\bar{X}_t = -A^\top \bar{X}_t \, \mathrm{d}t + C^\top \, \mathrm{d}\overleftarrow{\eta}_t + \frac{1}{2}\bar{\Sigma}_t B R^{-1} B^\top \bar{X}_t, \quad \bar{X}_T \sim N(0, P_f)$$

  where $\bar{\Sigma}_t = \mathsf{E}[\bar{X}_t \bar{X}_t^\top]$ and $\overleftarrow{\eta}$ is a backward Wiener process
- Using the Itö rule for backward process:

$$\frac{\mathrm{d}}{\mathrm{d}t}\bar{\Sigma}_t = -A^\top \bar{\Sigma}_t - \bar{\Sigma}_t A - C^\top C + \bar{\Sigma}_t B R^{-1} B^\top \bar{\Sigma}_t, \quad \Sigma_T = P_f$$

  Identical to Ricatti eq for P_t, therefore $\bar{\Sigma}_t = P_t$
- However, it can not be implemented using a simulator since $A^\top$ is not available.

EnKF for Control: Attempt II

- **Objective:** construct an EnKF to simulate the Ricatti eq.

$$\dot{P}_t = -A^\top P_t - P_t A - C^\top C + P_t B R^{-1} B^\top P_t, \quad P_T = P_f$$

- **Attempt II:** Define $S_t = P_t^{-1}$ which solves the eq.

$$\frac{\mathrm{d}}{\mathrm{d}t} S_t = A S_t + S_t A^\top - B R^{-1} B^\top + S_t C^\top C S_t, \quad S_T = P_T^{-1}$$

- Construct a stochastic process with variance equal to S_t

$$\mathrm{d}\bar{X}_t = A\bar{X}_t \,\mathrm{d}t + B\,\mathrm{d}\overleftarrow{\eta}_t + \frac{1}{2}\bar{\Sigma}_t C^\top C \bar{X}_t \,\mathrm{d}t, \quad \bar{X}_T \sim N(0, P_T^{-1})$$

- Using the Itö rule for backward process:

$$\frac{\mathrm{d}}{\mathrm{d}t}\bar{\Sigma}_t = A\bar{\Sigma}_t + \bar{\Sigma}_t A^\top - B R^{-1} B^\top + \bar{\Sigma}_t C^\top C \bar{\Sigma}_t, \quad \Sigma_T = S_T$$

  Identical to equation for S_t. Therefore, $P_t = S_t^{-1} = \bar{\Sigma}_t^{-1}$

- It can be implemented using a simulator

EnKF for Control: Attempt II

- **Objective:** construct an EnKF to simulate the Ricatti eq.

$$\dot{P}_t = -A^\top P_t - P_t A - C^\top C + P_t B R^{-1} B^\top P_t, \quad P_T = P_f$$

- **Attempt II:** Define $S_t = P_t^{-1}$ which solves the eq.

$$\frac{\mathrm{d}}{\mathrm{d}t} S_t = A S_t + S_t A^\top - B R^{-1} B^\top + S_t C^\top C S_t, \quad S_T = P_T^{-1}$$

- Construct a stochastic process with variance equal to S_t

$$\mathrm{d}\bar{X}_t = A\bar{X}_t \,\mathrm{d}t + B\,\mathrm{d}\overleftarrow{\eta}_t + \frac{1}{2}\bar{\Sigma}_t C^\top C \bar{X}_t \,\mathrm{d}t, \quad \bar{X}_T \sim N(0, P_T^{-1})$$

- Using the Itö rule for backward process:

$$\frac{\mathrm{d}}{\mathrm{d}t} \bar{\Sigma}_t = A\bar{\Sigma}_t + \bar{\Sigma}_t A^\top - B R^{-1} B^\top + \bar{\Sigma}_t C^\top C \bar{\Sigma}_t, \quad \Sigma_T = S_T$$

  Identical to equation for S_t. Therefore, $P_t = S_t^{-1} = \bar{\Sigma}_t^{-1}$

- It can be implemented using a simulator

EnKF for Control: Attempt II

- **Objective:** construct an EnKF to simulate the Ricatti eq.

$$\dot{P}_t = -A^\top P_t - P_t A - C^\top C + P_t B R^{-1} B^\top P_t, \quad P_T = P_f$$

- **Attempt II:** Define $S_t = P_t^{-1}$ which solves the eq.

$$\frac{\mathrm{d}}{\mathrm{d}t} S_t = A S_t + S_t A^\top - B R^{-1} B^\top + S_t C^\top C S_t, \quad S_T = P_T^{-1}$$

- Construct a stochastic process with variance equal to S_t

$$\mathrm{d}\bar{X}_t = A\bar{X}_t\,\mathrm{d}t + B\,\mathrm{d}\overleftarrow{\eta}_t + \frac{1}{2}\bar{\Sigma}_t C^\top C \bar{X}_t\,\mathrm{d}t, \quad \bar{X}_T \sim N(0, P_T^{-1})$$

- Using the Itö rule for backward process:

$$\frac{\mathrm{d}}{\mathrm{d}t}\bar{\Sigma}_t = A\bar{\Sigma}_t + \bar{\Sigma}_t A^\top - B R^{-1} B^\top + \bar{\Sigma}_t C^\top C \bar{\Sigma}_t, \quad \Sigma_T = S_T$$

Identical to equation for S_t. Therefore, $P_t = S_t^{-1} = \bar{\Sigma}_t^{-1}$

- It can be implemented using a simulator

EnKF for Control: Attempt II

- **Objective:** construct an EnKF to simulate the Ricatti eq.

$$\dot{P}_t = -A^\top P_t - P_t A - C^\top C + P_t B R^{-1} B^\top P_t, \quad P_T = P_f$$

- **Attempt II:** Define $S_t = P_t^{-1}$ which solves the eq.

$$\frac{\mathrm{d}}{\mathrm{d}t} S_t = A S_t + S_t A^\top - B R^{-1} B^\top + S_t C^\top C S_t, \quad S_T = P_T^{-1}$$

- Construct a stochastic process with variance equal to S_t

$$\mathrm{d}\bar{X}_t = A\bar{X}_t \,\mathrm{d}t + B\,\mathrm{d}\overleftarrow{\eta}_t + \frac{1}{2}\bar{\Sigma}_t C^\top C \bar{X}_t \,\mathrm{d}t, \quad \bar{X}_T \sim N(0, P_T^{-1})$$

- Using the Itö rule for backward process:

$$\frac{\mathrm{d}}{\mathrm{d}t}\bar{\Sigma}_t = A\bar{\Sigma}_t + \bar{\Sigma}_t A^\top - B R^{-1} B^\top + \bar{\Sigma}_t C^\top C \bar{\Sigma}_t, \quad \Sigma_T = S_T$$

Identical to equation for S_t. Therefore, $P_t = S_t^{-1} = \bar{\Sigma}_t^{-1}$

- It can be implemented using a simulator

EnKF for Control: Attempt II

- **Objective:** construct an EnKF to simulate the Ricatti eq.

$$\dot{P}_t = -A^\top P_t - P_t A - C^\top C + P_t B R^{-1} B^\top P_t, \quad P_T = P_f$$

- **Attempt II:** Define $S_t = P_t^{-1}$ which solves the eq.

$$\frac{\mathrm{d}}{\mathrm{d}t} S_t = A S_t + S_t A^\top - B R^{-1} B^\top + S_t C^\top C S_t, \quad S_T = P_T^{-1}$$

- Construct a stochastic process with variance equal to S_t

$$\mathrm{d}\bar{X}_t = A\bar{X}_t \,\mathrm{d}t + B \,\mathrm{d}\overleftarrow{\eta}_t + \frac{1}{2}\bar{\Sigma}_t C^\top C \bar{X}_t \,\mathrm{d}t, \quad \bar{X}_T \sim N(0, P_T^{-1})$$

- Using the Itö rule for backward process:

$$\frac{\mathrm{d}}{\mathrm{d}t} \bar{\Sigma}_t = A\bar{\Sigma}_t + \bar{\Sigma}_t A^\top - B R^{-1} B^\top + \bar{\Sigma}_t C^\top C \bar{\Sigma}_t, \quad \Sigma_T = S_T$$

  Identical to equation for S_t. Therefore, $P_t = S_t^{-1} = \bar{\Sigma}_t^{-1}$

- It can be implemented using a simulator

EnKF for control: Particle system

Particle system:

$$\mathrm{d}X_t^i = \underbrace{AX_t^i\,\mathrm{d}t + B\,\mathrm{d}\overleftarrow{\eta}_t^i}_{\text{dynamics simulator}} + \underbrace{\frac{1}{2}\Sigma_t^{(N)}C^\top(CX_t^i - 0)\,\mathrm{d}t}_{\text{RL correction}}, \quad X_T^i \overset{\text{i.i.d}}{\sim} \mathcal{N}(0, P_T^{-1})$$

where $\Sigma_t^{(N)} = \frac{1}{N}\sum_{i=1}^N X_t^i (X_t^i)^\top$

- It can be implemented using only a simulator $f(x,u) = Ax + Bu$
- The dynamics resembles the exploration and exploration terms
- It does not require initialization with stabilizing gain
- It can be used for both finite and infinite horizon settings

EnKF for control: Particle system

Particle system:

$$\mathrm{d}X_t^i = \underbrace{AX_t^i\,\mathrm{d}t + B\,\mathrm{d}\overleftarrow{\eta}_t^i}_{\text{dynamics simulator}} + \underbrace{\frac{1}{2}\Sigma_t^{(N)} C^\top (CX_t^i - 0)\,\mathrm{d}t}_{\text{RL correction}}, \quad X_T^i \overset{\text{i.i.d}}{\sim} \mathcal{N}(0, P_T^{-1})$$

where $\Sigma_t^{(N)} = \frac{1}{N}\sum_{i=1}^N X_t^i (X_t^i)^\top$

- It can be implemented using only a simulator $f(x,u) = Ax + Bu$
- The dynamics resembles the exploration and exploration terms
- It does not require initialization with stabilizing gain
- It can be used for both finite and infinite horizon settings

EnKF for control: Particle system

Particle system:

$$\mathrm{d}X_t^i = \underbrace{AX_t^i\,\mathrm{d}t + B\,\mathrm{d}\overleftarrow{\eta}_t^i}_{\text{dynamics simulator}} + \underbrace{\frac{1}{2}\Sigma_t^{(N)}C^\top(CX_t^i - 0)\,\mathrm{d}t}_{\text{RL correction}}, \quad X_T^i \overset{\text{i.i.d}}{\sim} \mathcal{N}(0, P_T^{-1})$$

where $\Sigma_t^{(N)} = \frac{1}{N}\sum_{i=1}^N X_t^i (X_t^i)^\top$

- It can be implemented using only a simulator $f(x,u) = Ax + Bu$
- The dynamics resembles the exploration and exploration terms
- It does not require initialization with stabilizing gain
- It can be used for both finite and infinite horizon settings

EnKF for control: Particle system

Particle system:

$$\mathrm{d}X_t^i = \underbrace{AX_t^i\,\mathrm{d}t + B\,\mathrm{d}\overleftarrow{\eta}_t^i}_{\text{dynamics simulator}} + \underbrace{\frac{1}{2}\Sigma_t^{(N)} C^\top (CX_t^i - 0)\,\mathrm{d}t}_{\text{RL correction}}, \quad X_T^i \overset{\text{i.i.d}}{\sim} \mathcal{N}(0, P_T^{-1})$$

where $\Sigma_t^{(N)} = \frac{1}{N}\sum_{i=1}^N X_t^i (X_t^i)^\top$

- It can be implemented using only a simulator $f(x,u) = Ax + Bu$
- The dynamics resembles the exploration and exploration terms
- It does not require initialization with stabilizing gain
- It can be used for both finite and infinite horizon settings

EnKF for control: Particle system

Particle system:

$$\mathrm{d}X_t^i = \underbrace{AX_t^i\,\mathrm{d}t + B\,\mathrm{d}\overleftarrow{\eta}_t^i}_{\text{dynamics simulator}} + \underbrace{\frac{1}{2}\Sigma_t^{(N)} C^\top (CX_t^i - 0)\,\mathrm{d}t}_{\text{RL correction}}, \quad X_T^i \overset{\text{i.i.d}}{\sim} \mathcal{N}(0, P_T^{-1})$$

where $\Sigma_t^{(N)} = \frac{1}{N}\sum_{i=1}^N X_t^i (X_t^i)^\top$

- It can be implemented using only a simulator $f(x,u) = Ax + Bu$
- The dynamics resembles the exploration and exploration terms
- It does not require initialization with stabilizing gain
- It can be used for both finite and infinite horizon settings

EnKF for control: summary

- Optimal control problem

$$\begin{aligned} \text{minimize} \quad & \int_0^T \left(\frac{1}{2}\|CX_t\|^2 + \frac{1}{2} U_t^\top R U_t \right) \mathrm{d}t + \frac{1}{2} X_T^\top P_f X_T \\ \text{subject to} \quad & \dot{X}_t = AX_t + BU_t =: f(X_t, U_t) \end{aligned}$$

- Exact solution: $U_t = -R^{-1}B^\top P_t X_t$ where P_t solves

$$\dot{P}_t = -A^\top P_t - P_t A - C^\top C + P_t B R^{-1} B^\top P_t, \quad P_T = P_f$$

- EnKF algorithm: approximate $P_t \approx \left(\frac{1}{N}\sum_{i=1}^N X_t^i (X_t^i)^\top\right)^{-1}$ where

$$\mathrm{d}X_t^i = \underbrace{AX_t^i \, \mathrm{d}t + BR^{\frac{1}{2}} \, \mathrm{d}\overleftarrow{\eta}_t^i}_{\text{dynamics simulator}} + \underbrace{\frac{1}{2}\Sigma_t^{(N)} C^\top (CX_t^i - 0) \, \mathrm{d}t}_{\text{RL correction}}, \quad X_T^i \overset{\text{i.i.d}}{\sim} \mathcal{N}(0, P_T^{-1})$$

EnKF for control: summary

- Optimal control problem

$$\begin{aligned} \text{minimize} \quad & \int_0^T \left(\frac{1}{2} \|CX_t\|^2 + \frac{1}{2} U_t^\top R U_t \right) \mathrm{d}t + \frac{1}{2} X_T^\top P_f X_T \\ \text{subject to} \quad & \dot{X}_t = AX_t + BU_t =: f(X_t, U_t) \end{aligned}$$

- Exact solution: $U_t = -R^{-1} B^\top P_t X_t$ where P_t solves

$$\dot{P}_t = -A^\top P_t - P_t A - C^\top C + P_t B R^{-1} B^\top P_t, \quad P_T = P_f$$

- EnKF algorithm: approximate $P_t \approx \left(\frac{1}{N} \sum_{i=1}^N X_t^i (X_t^i)^\top\right)^{-1}$ where

$$\mathrm{d}X_t^i = \underbrace{AX_t^i \,\mathrm{d}t + BR^{\frac{1}{2}} \,\mathrm{d}\overleftarrow{\eta}_t^i}_{\text{dynamics simulator}} + \underbrace{\frac{1}{2} \Sigma_t^{(N)} C^\top (CX_t^i - 0) \,\mathrm{d}t}_{\text{RL correction}}, \quad X_T^i \overset{\text{i.i.d}}{\sim} \mathcal{N}(0, P_T^{-1})$$

EnKF for control: summary

- Optimal control problem

$$\begin{aligned} \text{minimize} \quad & \int_0^T \left(\frac{1}{2}\|CX_t\|^2 + \frac{1}{2} U_t^\top R U_t \right) \mathrm{d}t + \frac{1}{2} X_T^\top P_f X_T \\ \text{subject to} \quad & \dot{X}_t = AX_t + BU_t =: f(X_t, U_t) \end{aligned}$$

- Exact solution: $U_t = -R^{-1}B^\top P_t X_t$ where P_t solves

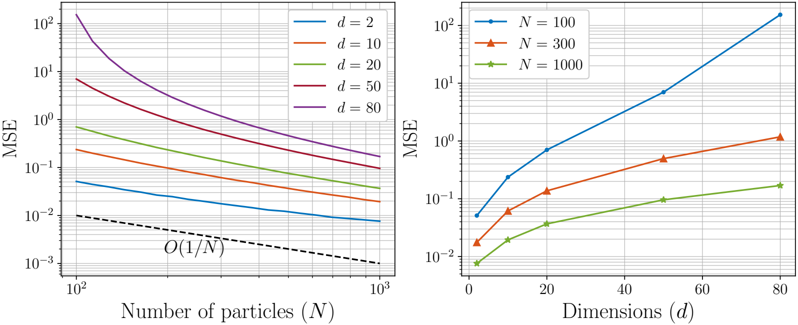
$$\dot{P}_t = -A^\top P_t - P_t A - C^\top C + P_t B R^{-1} B^\top P_t, \quad P_T = P_f$$

- EnKF algorithm: approximate $P_t \approx (\frac{1}{N}\sum_{i=1}^N X_t^i (X_t^i)^\top)^{-1}$ where

$$\mathrm{d}X_t^i = \underbrace{AX_t^i \,\mathrm{d}t + BR^{\frac{1}{2}} \,\mathrm{d}\overleftarrow{\eta}_t^i}_{\text{dynamics simulator}} + \underbrace{\frac{1}{2}\Sigma_t^{(N)} C^\top (CX_t^i - 0)\,\mathrm{d}t}_{\text{RL correction}}, \quad X_T^i \overset{\text{i.i.d}}{\sim} \mathcal{N}(0, P_T^{-1})$$

Outline

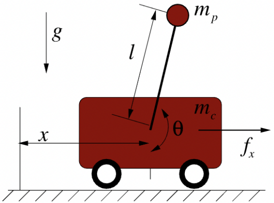
Numerical error analysis



Error as a function of number of particles N and the state dimension d

$$\text{error} = \mathsf{E}[\|K_{t_f}^{(N)} - K_{t_f}\|^2]$$

Cart-pole example



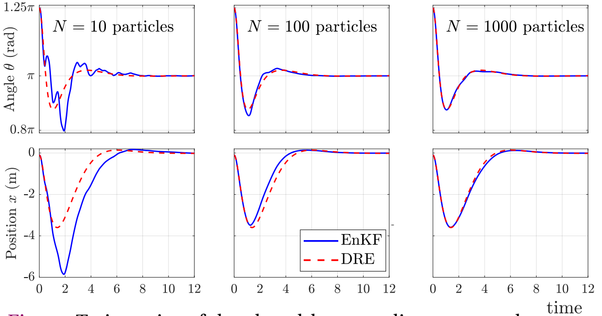


Figure: Trajectories of the closed-loop nonlinear cart pole system.

Outline

A nonlinear extension

- Optimal control problem:

$$\begin{aligned} \text{minimize} \quad & \mathsf{E}\left[\int_0^T \left(\frac{1}{2}c(X_t) + \frac{1}{2}U_t^2\right)\,\mathrm{d}t + g(X_T)\right] \\ \text{subject to} \quad & \mathrm{d}X_t = a(X_t)\,\mathrm{d}t + b(X_t)(U_t\,\mathrm{d}t + \mathrm{d}\xi_t) \end{aligned}$$

- This is a control affine system
- Each control channel is corrupted by additive Gaussian noise ξ_t
- Exact solution can be obtained using dynamic programming:

$$U_t = -b(X_t)\nabla v_t(X_t)$$

where $v_t(x)$ is the value function that solves the (nonlinear) HJB eq.

$$\frac{\partial v_t}{\partial t} + \mathcal{H}(v_t) = 0, \quad v_T = g$$

- Can we have an EnKF type algorithm to solve the HJB eq?

A nonlinear extension

- Optimal control problem:

$$\begin{aligned} \text{minimize} \quad & \mathsf{E}\left[\int_0^T \left(\frac{1}{2}c(X_t) + \frac{1}{2}U_t^2\right) \mathrm{d}t + g(X_T)\right] \\ \text{subject to} \quad & \mathrm{d}X_t = a(X_t)\,\mathrm{d}t + b(X_t)(U_t\,\mathrm{d}t + \mathrm{d}\xi_t) \end{aligned}$$

- This is a control affine system
- Each control channel is corrupted by additive Gaussian noise ξ_t
- Exact solution can be obtained using dynamic programming:

$$U_t = -b(X_t)\nabla v_t(X_t)$$

where $v_t(x)$ is the value function that solves the (nonlinear) HJB eq.

$$\frac{\partial v_t}{\partial t} + \mathcal{H}(v_t) = 0, \quad v_T = g$$

- Can we have an EnKF type algorithm to solve the HJB eq?

A nonlinear extension

- Optimal control problem:

$$\begin{aligned} \text{minimize} \quad & \mathsf{E}\left[\int_0^T \left(\frac{1}{2}c(X_t) + \frac{1}{2}U_t^2\right)\,\mathrm{d}t + g(X_T)\right] \\ \text{subject to} \quad & \mathrm{d}X_t = a(X_t)\,\mathrm{d}t + b(X_t)(U_t\,\mathrm{d}t + \mathrm{d}\xi_t) \end{aligned}$$

- This is a control affine system
- Each control channel is corrupted by additive Gaussian noise ξ_t
- Exact solution can be obtained using dynamic programming:

$$U_t = -b(X_t)\nabla v_t(X_t)$$

where $v_t(x)$ is the value function that solves the (nonlinear) HJB eq.

$$\frac{\partial v_t}{\partial t} + \mathcal{H}(v_t) = 0, \quad v_T = g$$

- Can we have an EnKF type algorithm to solve the HJB eq?

A nonlinear extension

- Optimal control problem:

$$\begin{aligned} \text{minimize} \quad & \mathsf{E}\left[\int_0^T \left(\frac{1}{2}c(X_t) + \frac{1}{2}U_t^2\right)\,\mathrm{d}t + g(X_T)\right] \\ \text{subject to} \quad & \mathrm{d}X_t = a(X_t)\,\mathrm{d}t + b(X_t)(U_t\,\mathrm{d}t + \,\mathrm{d}\xi_t) \end{aligned}$$

- This is a control affine system
- Each control channel is corrupted by additive Gaussian noise ξ_t
- Exact solution can be obtained using dynamic programming:

$$U_t = -b(X_t)\nabla v_t(X_t)$$

where $v_t(x)$ is the value function that solves the (nonlinear) HJB eq.

$$\frac{\partial v_t}{\partial t} + \mathcal{H}(v_t) = 0, \quad v_T = g$$

- Can we have an EnKF type algorithm to solve the HJB eq?

A nonlinear extension

- Optimal control problem:

$$\begin{aligned} \text{minimize} \quad & \mathsf{E}\left[\int_0^T \left(\frac{1}{2}c(X_t) + \frac{1}{2}U_t^2\right)\,\mathrm{d}t + g(X_T)\right] \\ \text{subject to} \quad & \mathrm{d}X_t = a(X_t)\,\mathrm{d}t + b(X_t)(U_t\,\mathrm{d}t + \,\mathrm{d}\xi_t) \end{aligned}$$

- This is a control affine system
- Each control channel is corrupted by additive Gaussian noise ξ_t
- Exact solution can be obtained using dynamic programming:

$$U_t = -b(X_t)\nabla v_t(X_t)$$

where $v_t(x)$ is the value function that solves the (nonlinear) HJB eq.

$$\frac{\partial v_t}{\partial t} + \mathcal{H}(v_t) = 0, \quad v_T = g$$

- Can we have an EnKF type algorithm to solve the HJB eq?

Log-transformation

- Define the probability $p_t(x) \propto e^{-v_t(x)}$ which solves the pde

$$\frac{\partial p_t}{\partial t} = -\underbrace{\nabla(p_t(a+\nabla b^2))}_{\text{drift term}} - \underbrace{\frac{1}{2}\nabla^2(p_t b^2)}_{\text{diffusion}} + p_t(h - \hat{h}_t), \quad p_T \propto e^{-g}$$

where $h := \frac{1}{2}c + \nabla a$ and $\hat{h}_t := \int h(x)p_t(x)\,\mathrm{d}x$

- Construct a stochastic process $\bar{X}_t$ so that $\mathrm{Law}(\bar{X}_t) = p_t$:

$$\mathrm{d}\bar{X}_t = a(\bar{X}_t)\,\mathrm{d}t + b(\bar{X}_t)\,\mathrm{d}\overleftarrow{\eta}_t + \nabla b^2(\bar{X}_t)\,\mathrm{d}t + \nabla\phi_t(\bar{X}_t)\,\mathrm{d}t$$

where $\overleftarrow{\eta}$ is backward Wiener process and ϕ_t solves the Poisson eq.

$$\text{Poisson eq.:} \quad -\frac{1}{\bar{p}_t(x)}\nabla\cdot(p_t(x)\nabla\phi_(x)t) = h(x) - \hat{h}_t$$

- Use $\bar{p}_t$ to evaluate the optimal control $U_t = b^\top(X_t)\nabla\log\bar{p}_t(X_t)$

Log-transformation

- Define the probability $p_t(x) \propto e^{-v_t(x)}$ which solves the pde

$$\frac{\partial p_t}{\partial t} = -\underbrace{\nabla(p_t(a + \nabla b^2))}_{\text{drift term}} - \underbrace{\frac{1}{2}\nabla^2(p_t b^2)}_{\text{diffusion}} + p_t(h - \hat{h}_t), \quad p_T \propto e^{-g}$$

where $h := \frac{1}{2}c + \nabla a$ and $\hat{h}_t := \int h(x)p_t(x)\,\mathrm{d}x$

- Construct a stochastic process $\bar{X}_t$ so that $\mathrm{Law}(\bar{X}_t) = p_t$:

$$\mathrm{d}\bar{X}_t = a(\bar{X}_t)\,\mathrm{d}t + b(\bar{X}_t)\,\mathrm{d}\overleftarrow{\eta}_t + \nabla b^2(\bar{X}_t)\,\mathrm{d}t + \nabla\phi_t(\bar{X}_t)\,\mathrm{d}t$$

where $\overleftarrow{\eta}$ is backward Wiener process and ϕ_t solves the Poisson eq.

$$\text{Poisson eq.:} \quad -\frac{1}{\bar{p}_t(x)}\nabla\cdot(p_t(x)\nabla\phi_(x)t) = h(x) - \hat{h}_t$$

- Use $\bar{p}_t$ to evaluate the optimal control $U_t = b^\top(X_t)\nabla\log\bar{p}_t(X_t)$

Log-transformation

- Define the probability $p_t(x) \propto e^{-v_t(x)}$ which solves the pde

$$\frac{\partial p_t}{\partial t} = -\underbrace{\nabla(p_t(a+\nabla b^2))}_{\text{drift term}} - \underbrace{\frac{1}{2}\nabla^2(p_t b^2)}_{\text{diffusion}} + p_t(h-\hat{h}_t), \quad p_T \propto e^{-g}$$

where $h := \frac{1}{2}c + \nabla a$ and $\hat{h}_t := \int h(x)p_t(x)\,\mathrm{d}x$

- Construct a stochastic process $\bar{X}_t$ so that $\mathrm{Law}(\bar{X}_t) = p_t$:

$$\mathrm{d}\bar{X}_t = a(\bar{X}_t)\,\mathrm{d}t + b(\bar{X}_t)\,\mathrm{d}\overleftarrow{\eta}_t + \nabla b^2(\bar{X}_t)\,\mathrm{d}t + \nabla\phi_t(\bar{X}_t)\,\mathrm{d}t$$

where $\overleftarrow{\eta}$ is backward Wiener process and ϕ_t solves the Poisson eq.

$$\text{Poisson eq.:} \quad -\frac{1}{\bar{p}_t(x)}\nabla\cdot(p_t(x)\nabla\phi_(x)t) = h(x) - \hat{h}_t$$

- Use $\bar{p}_t$ to evaluate the optimal control $U_t = b^\top(X_t)\nabla\log\bar{p}_t(X_t)$

Concluding remarks

- The nonlinear extension simplifies to EnKF in LQG setting
- The Poisson eq. also appears in nonlinear filtering algorithms (duality)
- The presented nonlinear extension is not completely simulation based (requires $\nabla \cdot a$ and ∇b^2)
- Can we use the controlled system particle framework to solve the filtering and control simultaneously?

Thanks for your attention!
Questions?

https://arxiv.org/abs/2107.01244

Concluding remarks

- The nonlinear extension simplifies to EnKF in LQG setting
- The Poisson eq. also appears in nonlinear filtering algorithms (duality)
- The presented nonlinear extension is not completely simulation based (requires $\nabla \cdot a$ and ∇b^2)
- Can we use the controlled system particle framework to solve the filtering and control simultaneously?

Thanks for your attention!
Questions?

https://arxiv.org/abs/2107.01244

Concluding remarks

- The nonlinear extension simplifies to EnKF in LQG setting
- The Poisson eq. also appears in nonlinear filtering algorithms (duality)
- The presented nonlinear extension is not completely simulation based (requires $\nabla \cdot a$ and ∇b^2)
- Can we use the controlled system particle framework to solve the filtering and control simultaneously?

Thanks for your attention!
Questions?

https://arxiv.org/abs/2107.01244

Concluding remarks

- The nonlinear extension simplifies to EnKF in LQG setting
- The Poisson eq. also appears in nonlinear filtering algorithms (duality)
- The presented nonlinear extension is not completely simulation based (requires $\nabla \cdot a$ and ∇b^2)
- Can we use the controlled system particle framework to solve the filtering and control simultaneously?

Thanks for your attention!
Questions?

https://arxiv.org/abs/2107.01244

Concluding remarks

- The nonlinear extension simplifies to EnKF in LQG setting
- The Poisson eq. also appears in nonlinear filtering algorithms (duality)
- The presented nonlinear extension is not completely simulation based (requires $\nabla \cdot a$ and ∇b^2)
- Can we use the controlled system particle framework to solve the filtering and control simultaneously?

Thanks for your attention!
Questions?

https://arxiv.org/abs/2107.01244

Concluding remarks

- The nonlinear extension simplifies to EnKF in LQG setting
- The Poisson eq. also appears in nonlinear filtering algorithms (duality)
- The presented nonlinear extension is not completely simulation based (requires $\nabla \cdot a$ and ∇b^2)
- Can we use the controlled system particle framework to solve the filtering and control simultaneously?

Thanks for your attention!
Questions?

https://arxiv.org/abs/2107.01244