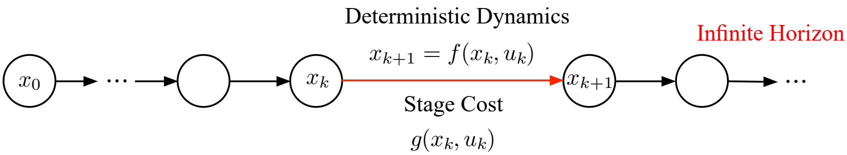
Data-driven Rollout for Deterministic Optimal Control

Yuchao Li[†], *Karl H. Johansson*[†], *Jonas Mårtensson*[†], *Dimitri P. Bertsekas*[‡]

[†]Division of Decision and Control Systems, KTH Royal Institute of Technology, Stockholm, Sweden

[‡]Fulton Professor of Computational Decision Making, ASU, Tempe, AZ, and McAfee Professor of Engineering, MIT, Cambridge, MA, USA

Introduction (1)



Problem of interest

- Problem data: $x \in X,\ u \in U(x) \subset U,\ x_{k+1} = f(x_k, u_k),\ g(x_k, u_k) \in [0, \infty]$.
- For every policy $\mu : X \to U$ so that $\mu(x) \in U(x)$ for all x, its *cost function* is

$$J_\mu(x_0) = \sum_{k=0}^{\infty} g\big(x_k, \mu(x_k)\big).$$

- The goal is to obtain the *optimal policy* μ^* such that J_{μ^*} equals the *optimal cost*:

$$J^*(x_0) = \min_{\substack{u_k \in U(x_k),\ k=0,1,\ldots \\ x_{k+1}=f(x_k,u_k),\ k=0,1,\ldots}} \sum_{k=0}^{\infty} g(x_k, u_k).$$

Introduction (2)

Approximation in value space

- Bellman's equation hold (cf. [Str66], [Ber15])

$$J^*(x) = \min_{u \in U(x)} \Big(g(x,u) + J^*\big(f(x,u)\big) \Big), \ \mu^*(x) \in \arg \min_{u \in U(x)} \Big(g(x,u) + J^*\big(f(x,u)\big) \Big).$$

- To overcome the *curse of dimensionality*, approximation in value space involves

$$\text{Off-line 'training': } \bar{J}(x); \ \text{On-line 'play': } \tilde{\mu}(x) \in \arg \min_{u \in U(x)} \Big(g(x,u) + \bar{J}\big(f(x,u)\big) \Big).$$

Rollout: Using the cost function J_μ of a base policy μ as $\bar{J}$

- Fundamental property: *sequential improvement condition* (introduced in [BTW97])

$$\min_{u \in U(x)} \Big\{ g(x,u) + J_\mu\big(f(x,u)\big) \Big\} \le J_\mu(x),$$

which hold regardless of the nature of state and control spaces, and dynamics.

Background: Theory

Theory on exact methods

- Related dynamic programming (DP) theory started with [Str66].
- Monotonicity property: If $J \leq \bar{J}$, then $g(x,u) + J\big(f(x,u)\big) \leq g(x,u) + \bar{J}\big(f(x,u)\big)$.
- Fixed point equations: For all policies μ, we have

$$J_\mu(x) = g\left(x, \mu(x)\right) + J_\mu\left(f\left(x, \mu(x)\right)\right), \; \forall x \in X.$$

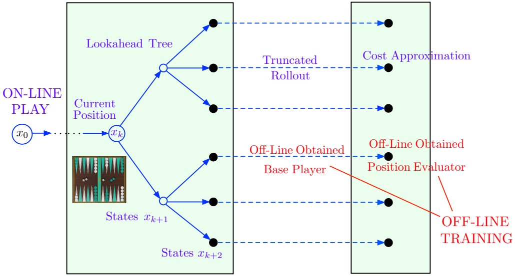
- Upper bounds: For all policies μ and some nonnegative function $\tilde{J} : X \to [0, \infty]$,

$$g\left(x, \mu(x)\right) + \tilde{J}\left(f\left(x, \mu(x)\right)\right) \leq \tilde{J}(x), \; \forall x \in X, \; \text{implies } J_\mu(x) \leq \tilde{J}(x), \; \forall x \in X.$$

Theory on computation

- When $\bar{J} = J_\mu$, the on-line 'play' step $\tilde{\mu}(x) \in \arg\min_{u \in U(x)} \Big(g(x,u) + \bar{J}\big(f(x,u)\big)\Big)$ is known to be one step of Newton's method with starting point J_μ for solving Bellman's equation (cf. [Kle68], [PoA69], [Hew71], [PuB79]).
- Even when approximation $\bar{J} \approx J_\mu$ is involved, similar interpretation of on-line 'play' as one step of Newton's or Newton-like method holds true ([Ber20], [Ber21])!

Background: Suboptimal Schemes (1)



Rollout and related methods

- $\bar{J} = J_\mu$ or $\bar{J} \approx J_\mu$, e.g., $\bar{J}(x_\ell) = \sum_{k=\ell}^{\ell+m-1} g(x_k, \mu(x_k)) + \hat{J}(x_{\ell+m})$, with base policy μ.
- One step lookahead: $\tilde{\mu}(x) \in \arg\min_{u \in U(x)} \Big(g(x,u) + \bar{J}\big(f(x,u)\big)\Big)$; or ℓ-step lookahead: $\tilde{\mu}(x_0) \in \arg\min_{u \in U(x_0)} \Big(g(x_0,u) + \min_{u_k,\, k=1,\ldots,\ell-1} \big(\sum_{k=1}^{\ell-1} g(x_k,u_k) + \bar{J}(x_\ell)\big)\Big)$
- Key theoretical concern: The rollout policy $\tilde{\mu}$ outperforms the base policy μ, e.g.,

$$J_{\tilde{\mu}}(x) \leq J_\mu(x), \text{ for all } x.$$

Background: Suboptimal Schemes (2)

$$\tilde{J}(x) = \underset{\{u_k\}_{k=0}^{\ell-1}}{\overset{\text{ON-LINE PLAY}}{\min}} \sum_{k=0}^{\ell+m-1} g(x_k, u_k) + \color{red}{G(x_{\ell+m})} \qquad \color{red}{\text{terminal cost}}$$

$$\begin{aligned} \text{s. t.} \quad & x_{k+1} = f(x_k, u_k),\ k = 0, ..., \ell + m - 1, \\ & x_k \in C,\ u_k \in U(x_k),\ \ k = 0, ..., \ell + m - 1, \\ & \color{red}{u_k = \mu(x_k),\ \ k = \ell, ..., \ell + m - 1,} \qquad \color{red}{\text{base policy}} \\ & \color{red}{x_{\ell+m} \in C_{\ell+m}} \qquad \color{red}{\text{terminal constraint}} \\ & x_0 = x. \end{aligned}$$

OFF-LINE TRAINING

Model predictive control (MPC)

- Off-line training: the terminal cost G, the terminal constraint $C_{\ell+m}$ & the base policy μ.
- On-line play: solving the numerical optimization problem.
- Key theoretical concerns:
  - Recursive feasibility of the numerical problem: related to μ and $C_{\ell+m}$
  - Stability of the closed loop system: using $\tilde{J}$ as Lyapunov function
- Close connections with DP and rollout (cf. [KeG88], [Ber05]).

Main results (1)

Basic form of data-driven rollout

- Exact rollout: $\bar{J} = J_\mu$, and we have $J_{\tilde{\mu}} \leq J_\mu$. However, we need to obtain the values of $J_\mu(x)$ for all $x \in X$.
- What if we can only compute $J_\mu(x)$ for $x \in S$, where $S \subset X$? Can we still get $J_{\tilde{\mu}} \leq J_\mu$?
- The answer is YES! But only for some S (key inspiration [RoB18]):

$$x \in S \implies f\big(x, \mu(x)\big) \in S.$$

- The effective $\bar{J}$ is given as

$$\bar{J}(x) = J_\mu(x) + \delta_S(x),$$

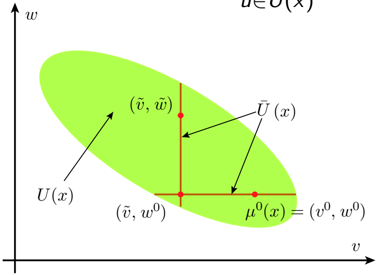
  where $\delta_S(\cdot)$ is an indicator function. This ensures the sequential improvement condition holds, which in turn guarantees that $J_{\tilde{\mu}} \leq J_\mu$. True for broad class of problems!
- The conventional Lyapunov function $\tilde{J}$ is an upper-bound of $J_{\tilde{\mu}}$; the recursive feasibility is implied by the sequential improvement condition.
- Enlarging the size of S improves the bound $\tilde{J}$, not necessarily the cost function $J_{\tilde{\mu}}$.

Main results (2)

Extensions

- If some constraint C_∞ is imposed on the entire trajectory $\{(x_k, u_k)\}_{k=0}^{\infty}$, then state augmentation (cf. [Ber20]) can be used, and the method remains valid.
- If there are multiple policies $\mu^0, \mu^1, \dots, \mu^n$ and multiple corresponding sets, a similar method applies (use the policy that is pointwise 'best').
- Attaining minimum of $\{g(x,u) + J_\mu(f(x,u))\}$ over $u \in U(x)$ is sufficient to ensure sequential improvement condition, but not necessary.

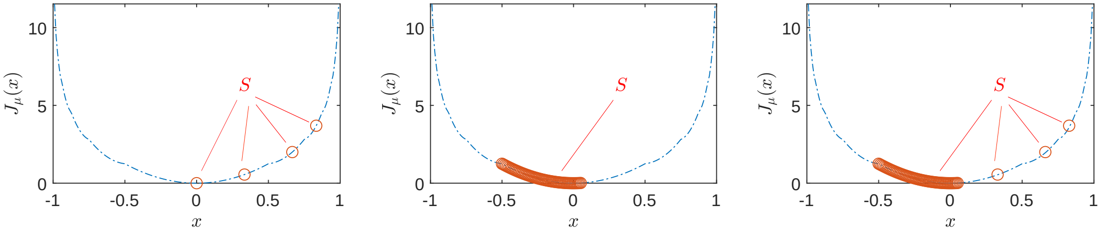
$$\text{If } \mu(x) \in \bar{U}(x) \subset U(x), \text{ then } \tilde{\mu}(x) \in \arg\min_{u \in \bar{U}(x)} \{g(x,u) + J_\mu(f(x,u))\} \leq J_\mu(x).$$



Illustrating example: The structure of the set S

A scalar linear quadratic problem

- Consider $X = (-1, 1)$, $U(x) = [-1, 1]$, $x_{k+1} = 2x_k + u_k$, and $g(x_k, u_k) = x_k^2 + u_k^2$.
- A base policy is given as $\mu(x) = -\text{sgn}(x)$ if $|x| > 1/2$ and $\mu(x) = -2x$ otherwise.
- Examples of possible set S: discrete points, continuous range, or a mixture!



- We can collect pieces of the cost function J_μ and assemble them together to form the set S, as long as the following condition is met:

$$x \in S \implies f\big(x, \mu(x)\big) \in S.$$

Conclusion

- We highlighted the similarities and conections between rollout and MPC.
- A data-driven variant of exact rollout is introduced, and the fixed point equation plays a central role for its analysis.
- The variant admits trajectory constrained, multiple policies and simplified extensions.
- A scalar linear quadratic regulation problem was used to illustrate the algorithm, while a few other examples are provided in [LJM21].

References (1)

[Ber05] Dimitri P. Bertsekas. "Dynamic programming and suboptimal control: A survey from ADP to MPC." *European Journal of Control*, 2005.

[Ber15] Dimitri P. Bertsekas. "Value and policy iterations in optimal control and adaptive dynamic programming," *IEEE Transactions on Neural Networks and Learning Systems*, 2015.

[Ber20] Dimitri P. Bertsekas. *Rollout, Policy Iteration, and Distributed Reinforcement Learning*, 2020.

[Ber21] Dimitri P. Bertsekas. "Lessons from alphazero for optimal, model predictive, and adaptive control." arXiv preprint arXiv:2108.10315 (2021).

[BTW97] Dimitri P. Bertsekas, John N. Tsitsiklis, and Cynara Wu. "Rollout algorithms for combinatorial optimization." *Journal of Heuristics*, 1997.

[Hew71] Gary Hewer. "An iterative technique for the computation of the steady state gains for the discrete optimal regulator," *IEEE Transactions on Automatic Control*, 1971.

[Kle68] David Kleinman. "On an iterative technique for Riccati equation computations," *IEEE Transactions on Automatic Control*, 1968.

References (2)

[LJM21] Yuchao Li, Karl H. Johansson, Jonas Mårtensson, Dimitri P. Bertsekas "Data-driven rollout for deterministic optimal control," IEEE CDC, 2021.

[MS71] Martin L. Puterman and Shelby L. Brumelle. "On the convergence of policy iteration in stationary dynamic programming," *Mathematics of Operations Research*, 1979.

[PoA69] M. A. Pollatschek, and B. Avi-Itzhak. "Algorithms for stochastic games with geometrical interpretation," *Management Science*, 1969.

[Str66] Ralph E. Strauch. "Negative dynamic programming," *The Annals of Mathematical Statistics*, 1966.

[RoB18] Ugo Rosolia, Francesco Borrelli. "Learning model predictive control for iterative tasks: A Data-Driven Control Framework," *IEEE Transactions on Automatic Control*, 2018.