Adaptive optimal control of continuous-time nonlinear affine systems via hybrid iteration - NOTES

Devesh

December 1, 2024

1 Problem Formulation

Value Iteration is slow to converge.

Policy Iteration is prone to numerical instability.

Consider the nonlinear affine system given by:

$$\dot{x} = f(x) + g(x)u, \quad x(0) = x_0$$

The cost functional is defined as:

$$J(x, u) = \int_0^\infty \left[Q(x) + u^T R u\right] dt,$$

where $Q(x)$ is a positive semi-definite state cost and R is a positive definite control cost matrix.

The optimal value function is defined as:

$$V^*(x_0) = \inf_u J(x_0; u), \quad u^* = \arg\inf_u J(x_0; u)$$

The optimal value function V^* solves the Hamilton–Jacobi–Bellman (HJB) equation:

$$\inf_u H(x, \nabla_x V, u) = 0, \quad V(0) = 0, \quad \forall x \in \mathbb{R}^n,$$

The Hamiltonian is defined as:

$$H(x, \nabla_x V, u) := (\nabla_x V)^T (f(x) + g(x)u) + Q(x) + u^T R u,$$

The term $(\nabla_x V)^T (f(x) + g(x)u)$ represents the rate of change of V along the system's trajectory. It indicates how the value function $V(x)$ changes in the direction of the vector $f(x) + g(x)u$. Since $\nabla_x V(x)$ points in the direction of the steepest ascent of $V(x)$, the entire expression is a scalar that quantifies this rate of change.

Q(x) is the control cost and $u^T Ru$ is the state cost.

The HJB equation can be written as:

$$0 = (\nabla_x V^*)^T f(x) + Q(x) - \frac{1}{4}(\nabla_x V^*)^T g(x) R^{-1} g(x)^T \nabla_x V^*,$$

The optimal control law is given by:

$$u^*(x) = -\frac{1}{2} R^{-1} g(x)^T \nabla_x V^*, \quad \forall x \in \mathbb{R}^n.$$

2 Model Based Policy Iteration

Policy Evaluation: Using $u_i \in \mathcal{D}$, solve V_i from

$$H(x, \nabla_x V_i, u_i) = 0, \quad V_i(0) = 0.$$

$\mathcal{D}$ is the set of admissible control inputs.

Policy Improvement: Update the control policy by

$$u_{i+1} = -\frac{1}{2} R^{-1} g(x)^T \nabla_x V_i.$$

3 Model Based Value Iteration

Value Update: Solve $V_i(x)$ from

$$\inf_u \{H(x, \nabla_x V_i, u_{i+1})\} = 0,$$

where u_{i+1} is defined in.

In practice, can be solved using different numerical methods, such as stochastic approximation and forward Euler method.

$$V_{i+1}(x) \leftarrow V_i(x) + \epsilon_i \left((\nabla_x V_i(x))^T f(x) + Q(x) - (u_{i+1}(x))^T R u_{i+1}(x) \right)$$

where the sequence $\{\epsilon_i\}_{i=0}^{\infty}$ is a deterministic sequence. .

.

.

.

.

.

4 Model Based Hybrid Iteration

Algorithm 1 Model-based Hybrid Iteration

Choose a proper $V_0 \in \mathcal{P}$, $V_0(0) = 0$, and $\hat{Q}(x) \succ Q(x)$.
$i \leftarrow 0$
repeat
    Compute $u_{i+1} = -\frac{1}{2}R^{-1}(g(x))^T \nabla_x V_i(x)$.
    Update the value function using

$$V_{i+1}(x) \leftarrow V_i(x) + \epsilon_i \Big((\nabla_x V_i(x))^T f(x) + \hat{Q}(x) - (u_{i+1}(x))^T R u_{i+1}(x) \Big) .$$

    $i \leftarrow i + 1$
until $V_i(x) - V_{i-1}(x) \preceq \epsilon_{i-1}[\hat{Q}(x) - Q(x)]$ and $V_{i-1} \in \mathcal{P}^+$.
loop
    Solve V_i from

$$H(x, \nabla_x V_i, u_i) = 0.$$

    Update the control policy by

$$u_{i+1} = -\frac{1}{2}R^{-1}(g(x))^T \nabla_x V_i(x).$$

    $i \leftarrow i + 1$
end loop

Phase 1: This warm-up phase learns a baseline control policy ensuring system stability and cost-effectiveness. The stopping condition prevents excessive refinement of the value function. The criterion $V_i(x) - V_{i-1}(x) \preceq \epsilon_{i-1}[\hat{Q}(x) - Q(x)]$ ensures diminishing updates and penalizes suboptimal states via $\hat{Q}(x)$.

It ensures that the control policy $u_i(x)$ becomes admissible (stabilizing and finite-cost). It establishes a proper value function $V(x)$, which corresponds to the learned admissible policy.

Phase 2: This phase refines the control policy by solving the HJB equation, focusing on precision whereas in Phase 1 $V(x)$ was calculated using relaxed constraints $\hat{Q}$. The value function is updated to reflect the new control policy. The process iterates until convergence.