Stackelberg ILQGames Writeup

Hamzah Khan

November 13, 2022

In *Efficient Iterative Linear-Quadratic Approximations for Nonlinear Multi-Player General-Sum Differential Games*, Fridovich-Keil et al. 2020 introduce the ILQGames algorithm. ILQGames iteratively solves N-player games which do not satisfy linear-quadratic (LQ) assumptions on game dynamics and player costs.

In this document, we describe a similar algorithm, called 2-Player Stackelberg ILQGames, which adapts the original ILQGames algorithm to iteratively solve non-linear, non-quadratic 2-player Stackelberg games.

We assume no affine terms in this initial draft. TODO: adjust derivation to include affine terms.

1 Steps

The algorithm accepts a number of inputs:

- initial state $\boldsymbol{x}_0$,
- a horizon T over which the game is played,
- cost functions g_t^i, $i \in \{1, 2\}$ at each time step t,
- initial control strategies $\gamma^i_{k=0}$ described by the control gain matrix $S^i_{t,k=0}$ and the recursive cost matrix $L^i_{t,k=0}$, $i \in \{1, 2\}$.

We pre-select arbitrary reference state and controls to be

$$\boldsymbol{x}_t = 0,\ \boldsymbol{u}_t^i = 0 \quad \forall i \in \{1, 2\}$$

and $\eta \in (0, 1]$.

At each iteration $k = 1, 2, \ldots$ until convergence condition $\|\Delta\gamma_k^i\| <$ threshold ϵ,

1. Propagate γ^i_{k-1} forward in time using the gain matrices $S^i_{1:T,k=i}$ from $\boldsymbol{x}_0$ to get $\xi_k \equiv \{\hat{\boldsymbol{x}}_{1:T}, \hat{\boldsymbol{u}}^i_{1:T}\}$.
2. Compute errors
$$\delta\boldsymbol{x}_t = \boldsymbol{x}_t - \hat{\boldsymbol{x}}_{1:T}, \quad \delta\boldsymbol{u}_t^i = \boldsymbol{u}_t^i - \hat{\boldsymbol{u}}^i_{1:T}.$$
3. Linearize the dynamics and quadraticize about the error state and controls.
$$A_t = \frac{\partial f_t(\hat{\boldsymbol{x}}_t, \hat{\boldsymbol{u}}_t^i)}{\partial \hat{\boldsymbol{x}}_t}, \quad B_t^i = \frac{\partial f_t(\hat{\boldsymbol{x}}_t, \hat{\boldsymbol{u}}_t^i)}{\partial \hat{\boldsymbol{u}}_t^i}$$
$$Q_t^i = \nabla_{\hat{\boldsymbol{x}}\hat{\boldsymbol{x}}} g_t^i, \quad R_t^{ij} = \nabla_{\hat{\boldsymbol{u}}^j\hat{\boldsymbol{u}}^j} g_t^i$$

4. Solve LQ Stackelberg game $\gamma_k^i = \mathcal{S}_t^1(A_t, B_t, \{Q_t^i\}, \{R_t^{ij}\}, T; \boldsymbol{x}_0) = \left\{S_{1:T,k}^i, L_{1:T,k}^i\right\}$.
5. Adjust the control strategies based on this approximate solution:

$$\gamma_k^i = \hat{\boldsymbol{u}}_{1:T}^i - S_{1:T,k}^i \delta \boldsymbol{x}_t.$$