CSC_5RO01_TA

Hybrid Optimal Control

Lecture 4: Differential Games

Mario Gleirscher (v0.1, 12/12/25)

3A/Master Course, 2 ETCS, 2025/26

Learning Goals

Questions:

1. How can we identify an optimal controller for a system subject to disturbance?
2. How can we switch between different representations of a hybrid system?

At the end of this lecture, you will ...

1. have learned about differential games as a basis for the synthesis of robust controllers for hybrid systems and
2. have seen a recipe to convert hybrid automata (HAs) systematically into hybrid programs (HPs).

How can we compute a controller that is optimal under disturbance?

Closed-Loop Zero-Sum Differential Games

We use a single cost functional

$$J(u_1, u_2) = \int_{t_0}^{t_f} L(u_1, u_2; x(\tau), \tau)\, \mathrm{d}\tau + \phi(x; t_f)$$

and require $J = J_1 = -J_2$. Let $u_i : \mathbb{R}_{\geqslant 0} \times \mathcal{X} \to \mathcal{U}_i$ for $i \in 1, 2$.

Definition (Saddle Point as a Nash Equilibrium)

The pair of feedback controllers (u_1^*, u_2^*) forms a saddle point if

$$\forall u_1, u_2 \colon\ J(u_1^*, u_2) \leqslant \underbrace{J(u_1^*, u_2^*)}_{\text{value of the game}} \leqslant J(u_1, u_2^*)$$

where

$$u_1^*(x; t) = \arg\min_{u_1} \max_{u_2} \left\{ L(u_1, u_2; x(t), t) + \frac{\partial V(x; t)}{\partial x} f(u_1, u_2; x(t), t) \right\}$$

$$u_2^*(x; t) = \arg\max_{u_2} \min_{u_1} \left\{ L(u_1, u_2; x(t), t) + \frac{\partial V(x; t)}{\partial x} f(u_1, u_2; x(t), t) \right\} .$$

Solving Closed-Loop Zero-Sum DGs (via Continuous DP)

Solve the Hamilton-Jacobi-Isaacs (HJI) equation for V for two players $i \in 1, 2$,

$$-\frac{\partial V(x;t)}{\partial t} = \min_{u_1} \max_{u_2} \left\{ L(u_1, u_2; x(t), t) + \frac{\partial V(x;t)}{\partial x} f(u_1, u_2; x(t), t) \right\}$$
$$= \max_{u_2} \min_{u_1} \left\{ L(u_1, u_2; x(t), t) + \frac{\partial V(x;t)}{\partial x} f(u_1, u_2; x(t), t) \right\},$$

for $t \in [t_0, t_f]$ with boundary condition $V(x_f, t_f) = \phi(x_f, t_f)$.

Solving this partial differential equation for V yields the value of the game $V(x_0, t_0)$.

(Lin et al. 2022, p. 340)

Discrete DP of a Differential Game by Value Iteration

Idea: Derive discrete dynamics $x_{k+1} = f(x_k, u_k, d_k, k)$ and solve the Hamilton-Jacobi-Bellman (HJB) equation offline by recursion of the value function.

procedure DDP(in $\mathcal{X}$, $(f, \mathcal{U}, \mathcal{D})$, $J[L, \Phi]$, $[t_0, t_\mathrm{f}]_{\delta t}$; out V, $\mathbf{u}^*$)

- $(V, \mathbf{u}^*, d^*) \leftarrow \perp^{|\mathcal{X}| \times [t_0, t_\mathrm{f}]}$ ▷ $\perp$ undef., e.g., $\mathbf{0}$
- $V(\mathcal{X}, t_\mathrm{f}) \leftarrow \Phi(\mathcal{X})$
- **for** $k \in [t_0, t_\mathrm{f} - \delta t]$ **do** ▷ backwards
  - **for** $x_k \in \mathcal{X}$ **do**
    - $\mathcal{X}_{k+\delta t}^{\mathcal{U}_k, \mathcal{D}_k} \leftarrow x_k + f'(x_k, \mathcal{U}_k, \mathcal{D}_k, k)\delta t$ ▷ forward Euler set
    - $V(x_k, k) \leftarrow \min_{u_k \in \mathcal{U}_k} \max_{d_k \in \mathcal{D}_k} \left\{ L(x_k, u_k, d_k, k) + V(x_{k+\delta t}^{u_k, d_k}, k + \delta t) \right\}$
    - $\mathbf{u}^*(x_k, k) \leftarrow \arg \min_{u_k \in \mathcal{U}_k} \max_{d_k \in \mathcal{D}_k} \left\{ L(x_k, u_k, d_k, k) + V(x_{k+\delta t}^{u_k, d_k}, k + \delta t) \right\}$

for discretised states $\mathcal{X}$, inputs $\mathcal{U}$, disturbances $\mathcal{D}$, and finite horizon $[t_0, t_\mathrm{f}]_{\delta t}$ and with $x_{k+\delta t}^{u_k, d_k} \in \mathcal{X}_{k+\delta t}^{\mathcal{U}_k, \mathcal{D}_k}$.

Excursion: How can we switch between different representations of a hybrid system?

Translation of Hybrid Automata into Hybrid Programs

Hybrid Systems Formalisms (recap)

A hybrid automaton (HA) is a tuple $H = (Gra, X, Ini, Inv, f, Jmp, F)$ with $Gra = (Q, A, E)$, a set of modes Q, an alphabet A, and a set of events $E \subseteq Q \times A \times Q$.

A hybrid program (HP) is a regular expression H formed by composing atoms (i.e., discrete assignments $\mathbf{x} := e$, continuous assignments $\dot{\mathbf{x}} = f \& \chi$, conditions χ) using sequence ($H ; H$), choice ($H \cup H$), test ($?\chi$), and iteration ($H^\star$).

1 From each mode (or location) $l \in Q$ and all its outgoing transitions $(l, e, l') \in E$, derive a HP L of the form

$$L \equiv f(l); \bigcup_{(l,e,l') \in E} ?\mathrm{grd}(e); \mathrm{upd}(e); l' .$$

If l is the initial mode of H, define $H' \equiv L$.

Translation of Hybrid Automata into Hybrid Programs

❷ Remove all occurrences of recursion in H'.

Problem: (mutual) recursion after direct translation of HA
Solution: replace (mutual) recursion by iteration

Examples:

1. Mutual recursion, starting from A:

$$A := a; B$$
$$B := b; A$$

$$A' := a; b; A'$$

- $A' := (a; b)^\star$

2. Recursion with branching:

$$B := b; B \cup c; A$$

$$B' := b^\star; c; A$$

- $B' := b^\star; c; A$

3. Mutual recursion with branching, starting from A:

$$A := a; B$$
$$B := b; B \cup c; A$$

$$B' := b^\star; c; A$$
$$A' := a; b^\star; c; A'$$

- $A' := (a; b^\star; c)^\star$

with hybrid programs a, b, c and identifiers A, B, A', B'.

Translation of Hybrid Automata into Lince Hybrid Programs

Translate a non-recursive HP H' into a Lince hybrid program:

3. Convert each non-deterministic iteration $H^\star$ into `while true` $\{\, H \,\}$.

4. For a terminating program, refine `true` with a finite bound n on a monotonic counter, for example, $i := 0$; `while` $i < n \; \{\, H; i{+}{+} \,\}$.

5. Refine each non-deterministic guarded choice, that is, an expression of the form

$$(?\mathrm{grd}(e_1); H_1) \cup \cdots \cup (?\mathrm{grd}(e_n); H_n)$$

into an ordered conditional choice

$$\begin{aligned}&\texttt{if }(\mathrm{grd}(e_1))\texttt{ then } H_1 \quad \ldots\\ &\texttt{else if }(\mathrm{grd}(e_n))\texttt{ then } H_n\\ &\texttt{else } \mathit{Err}\end{aligned}$$

(Err...a suitable exception handler)

Note how $?c; H$ leads to `if` c `then` $\{H\}$ `else` $\{\mathit{Err}\}$.

6. Convert restricted continuous assignments $\dot{\mathbf{x}} = f \& \chi$ into $\dot{\mathbf{x}} = f$ `until_`$d\ \neg\chi$, where $d \in \mathbb{R}_{\geqslant 0}$ is an appropriate evolution period.

Summary

- At a Nash equilibrium, neither player i can unilaterally gain anything by diverging from $u_i^\star$. A Nash equilibrium forms a solution to a cost- and/or dynamics-coupled optimal control problem.
- Discretised versions of zero-sum differential games can be solved by value iteration.
- With value iteration for DP, optimal controllers can be computed for non-linear systems and cost functionals, though at cost exponential in the dimensions of the state and input spaces.
- HAs and HPs provide graphical and language-theoretic views on the structure of a complex control problem.
- Modelling at scale can be achieved by, for example,, hierarchical structuring of HAs or using iteration and recursion in HPs.

References I

Lin, Hai and Panos J. Antsaklis (2022). Hybrid Dynamical Systems. Springer.