PACMAN^{HJ}

A classical arcade videogame powered by Hamilton-Jacobi equations

Simone Cacace

Università degli Studi Roma Tre

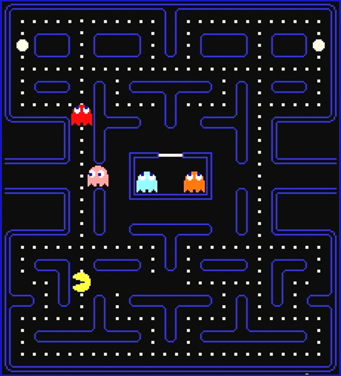
Control of State Constrained Dynamical Systems

25-29 September 2017, Padova

Main Menu

- Back to the 80's: the original game
- Optimal Control and Differential Games
- Semi-Lagrangian solvers for Hamilton-Jacobi equations
- Pacman^{HJ}
  - (Some) Implementation details
  - Play for your pill: solving the HJ-Bellman equation
  - Play for your life: solving the HJ-Isaacs equation
  - Modelling power pills via Hybrid Systems
  - The full game... an open problem!

Back to the 80's: the original game (Tohru Iwatani)



- Goal: eat all the pills avoiding ghosts
- Power pills make ghosts temporary harmless
- Predetermined strategies for ghosts (GET method)
- No cooperation

Optimal Control

- **Controlled Dynamical System**

$$\begin{cases} \dot{y}(t) = f(y(t), \alpha(t)), & t > 0 \\ y(0) = x \end{cases} \dashrightarrow y_x(t; \alpha)$$

- **State** $y(t) \in \mathbb{R}^n$
- **Admissible compact control set** $A \subset \mathbb{R}^m$
- **Dynamics** $f : \mathbb{R}^n \times A \to \mathbb{R}^n$ **(Lipschitz continuous)**
- **Control** $\alpha \in \mathcal{A} = \{$**measurable** $\alpha : [0, +\infty) \to A\}$
- $y_x(t; \alpha)$ **trajectory starting from** $x \in \mathbb{R}^n$ **using** $\alpha \in \mathcal{A}$

Find an optimal control $\alpha^* \in \mathcal{A}$ **minimizing some cost** $J(x, \alpha)$ **among all the trajectories starting from** x**.**

The Minimum Time Problem

- **Target** $\mathcal{T} \subset \mathbb{R}^n$ **closed set**
- **First arrival time** $J(x, \alpha) = \inf \{t \geq 0 \, : \, y_x(t; \alpha) \in \mathcal{T}\}$
- **Value function** $T(x) = \inf_{\alpha \in \mathcal{A}} J(x, \alpha)$
- **Dynamic programming principle**

$$T(x) = \inf_{\alpha \in \mathcal{A}} \{t + T(y_x(t; \alpha))\} \qquad \textbf{for all } 0 \leq t \leq T(x)$$

- **Hamilton-Jacobi-Bellman equation**

$$\begin{cases} \max_{a \in A} \{-f(x, a) \cdot \nabla T(x)\} = 1 & x \in \mathbb{R}^n \setminus \mathcal{T} \\ T(x) = 0 & x \in \mathcal{T} \end{cases}$$

- **Feedback optimal controls** $a^*(x) := \arg\max_{a \in A} \{-f(x, a) \cdot \nabla T(x)\}$
- **Optimal trajectories**

$$\begin{cases} \dot{y}^*(t) = f(y^*(t), a^*(y^*(t))) & 0 \leq t \leq T(x) \\ y^*(0) = x \end{cases}$$

Zero-sum Differential Games

- **Controlled Dynamical System**

$$\begin{cases} \dot{y}(t) = f(y(t), \alpha(t), \beta(t)), & t > 0 \\ y(0) = x \end{cases} \dashrightarrow y_x(t; \alpha, \beta)$$

- **State** $y(t) \in \mathbb{R}^n$
- **Admissible compact control sets** $A \subset \mathbb{R}^{m_A}$, $B \subset \mathbb{R}^{m_B}$
- **Dynamics** $f : \mathbb{R}^n \times A \times B \to \mathbb{R}^n$ **(Lipschitz continuous)**
- **Controls**

$$\alpha \in \mathcal{A} = \{\textbf{measurable } \alpha : [0, +\infty) \to A\}$$

$$\beta \in \mathcal{B} = \{\textbf{measurable } \beta : [0, +\infty) \to B\}$$

- $y_x(t; \alpha, \beta)$ **trajectory starting from** $x \in \mathbb{R}^n$ **using** $\alpha \in \mathcal{A}$ **and** $\beta \in \mathcal{B}$

Minimax solutions: find strategies $\alpha^* \in \mathcal{A}$, $\beta^* \in \mathcal{B}$ **such that**

$$J(x, \alpha^*, \beta) \leq J(x, \alpha^*, \beta^*) \leq J(x, \alpha, \beta^*)$$

Pursuit-Evasion Games

- **Dynamics** $f(x, a, b) := (f_p(x_p, a), f_e(x_e, b))$, $x = (x_p, x_e) \in \mathbb{R}^{2n}$
- **Target** $\mathcal{T} = \{z = (z_p, z_e) \in \mathbb{R}^{2n} \; : \; |z_p - z_e| \le \varepsilon\}$
- **Capture time** $J(x, \alpha, \beta) = \inf\{t \ge 0 \; : \; y_x(t; \alpha, \beta) \in \mathcal{T}\}$
- **Non anticipating strategies**

$$\Delta_{\mathcal{A}} := \left\{ \begin{array}{c} \Phi : \mathcal{B} \to \mathcal{A} \quad : \quad \forall t > 0, \quad \forall \beta, \tilde{\beta} \in \mathcal{B} \\ \beta(s) = \tilde{\beta}(s) \;\; \forall s \le t \quad \Rightarrow \quad \Phi[\beta](s) = \Phi[\tilde{\beta}](s) \;\; \forall s \le t \end{array} \right\}$$

$$\Delta_{\mathcal{B}} := \left\{ \begin{array}{c} \Psi : \mathcal{A} \to \mathcal{B} \quad : \quad \forall t > 0, \quad \forall \alpha, \tilde{\alpha} \in \mathcal{A} \\ \alpha(s) = \tilde{\alpha}(s) \;\; \forall s \le t \quad \Rightarrow \quad \Psi[\alpha](s) = \Psi[\tilde{\alpha}](s) \;\; \forall s \le t \end{array} \right\}$$

- **Value function (Isaacs conditions)**

$$T(x) = \inf_{\Phi \in \Delta_{\mathcal{A}}} \sup_{\beta \in \mathcal{B}} J(x, \Phi[\beta], \beta) = \sup_{\Psi \in \Delta_{\mathcal{B}}} \inf_{\alpha \in \mathcal{A}} J(x, \alpha, \Psi[\alpha])$$

Pursuit-Evasion Games

- **Dynamic programming principle**

$$T(x) = \inf_{\Phi \in \Delta_{\mathcal{A}}} \sup_{\beta \in \mathcal{B}} \{ t + T(y_x(t; \Phi[\beta], \beta)) \} \quad \textbf{for all } 0 \leq t \leq T(x)$$

- **Hamilton-Jacobi-Isaacs equation**

$$\begin{cases} \max_{a \in A} \min_{b \in B} \{ -f(x,a,b) \cdot \nabla T(x) \} = 1 & x \in \mathbb{R}^{2n} \setminus \mathcal{T} \\ T(x) = 0 & x \in \mathcal{T} \end{cases}$$

- **Optimal strategies**

$$(a^*(x), b^*(x)) := \arg \max_{a \in A} \min_{b \in B} \{ -f(x,a,b) \cdot \nabla T(x) \}$$

- **Optimal trajectories**

$$\begin{cases} \dot{y}_p^*(t) = f_p(y_p^*(t), a^*(y_p^*(t), y_e^*(t))) & y_p(0) = x_p & x \in \mathbb{R}^{2d} \setminus \mathcal{T} \\ \dot{y}_e^*(t) = f_e(y_e^*(t), b^*(y_p^*(t), y_e^*(t))) & y_e(0) = x_e & 0 \leq t \leq T(x) \end{cases}$$

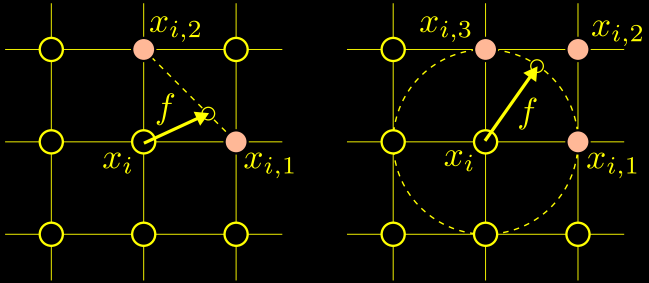
Semi-Lagrangian solvers for HJ equations

Discretize the HJ equation (or the DPP) on a Δx-grid:

- **ODE solver for the underlying dynamical system**
- **Interpolation for the reconstruction of non-grid values**

Example: Forward Euler + Linear Interpolation

- **Approximated trajectory** $y_{x_i}(\Delta t; a, b) \sim x_i + \Delta t\, f(x_i, a, b)$
- **Fixed point scheme** $T(x_i) = \max_{a \in A} \min_{b \in B} \{\Delta t + \mathcal{I}[T](x_i + \Delta t\, f(x_i, a, b))\}$



Pacman^{HJ}: (some) implementation details

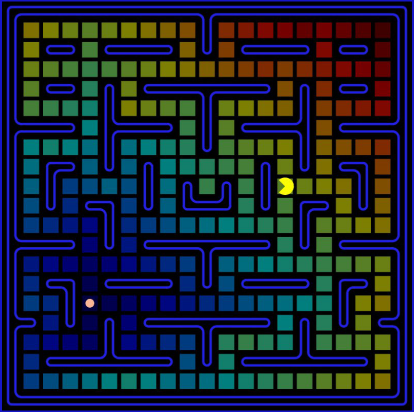
- **Pursuit-Evasion Game in a 2D Maze**
- **2 pursuers (Ghosts) and 1 evader (Pacman)**
- **Uniform 21×21 grid $\mathcal{G}$ with space step $\Delta x = 1$**
- $\mathcal{G} = \mathcal{M} \cup \mathcal{O}$, **Maze $\mathcal{M}$ (224 nodes), Obstacle $\mathcal{O}$ (217 nodes)**
- **Control set $\mathcal{C} = \{0, N, S, W, E\} \subset \mathbb{R}^2$**
- **Connectivity map $\mathcal{L} : \mathcal{M} \mapsto \mathcal{C}$ (embeds state/control constraints)**
- **Dynamics $f(x, a) = a \in \mathcal{L}(x)$ for all players**
- **Time step $\Delta t = 1 \Rightarrow \mathcal{M}$ is invariant under $f \Rightarrow$ No interpolation**
- **Coding only with integers in exact arithmetic**
- **Value functions precomputed and stored in a few Mbyte binary file**
- **Optimal strategies computed in real-time**

Pacman$^{\text{HJ}}$: play for your pill

- Solving the HJ-Bellman equation for each pill $k \in \{0, ..., 223\} \equiv \mathcal{M}$

$$\begin{cases} T^k_{\mathbf{HJB}}(s) = \min\limits_{\bar{s} \in \mathcal{N}_s} \left\{ T^k_{\mathbf{HJB}}(\bar{s}) + 1 \right\} & \text{if } s \neq k \\ T^k_{\mathbf{HJB}}(s) = 0 & \text{if } s = k \end{cases}$$

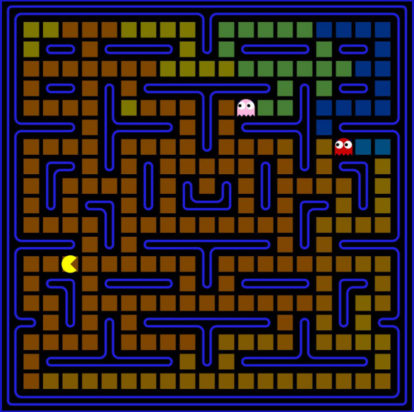
$\mathcal{N}_s$ is the set of indices of admissible (through $\mathcal{L}$) neighbors of s



Pacman$^{\mathbf{HJ}}$: play for your life

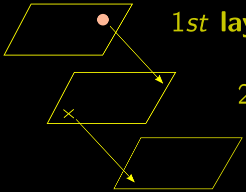
- **Solving the HJ-Isaacs equation for each** $(p_1, p_2, e) \in \mathcal{M}$

$$\begin{cases} T_{\mathbf{HJI}}(p_1, p_2, e) = \max\limits_{\overline{e} \in \mathcal{N}_e} \min\limits_{\substack{\overline{p}_1 \in \mathcal{N}_{p_1} \\ \overline{p}_2 \in \mathcal{N}_{p_2}}} \left\{ T_{\mathbf{HJI}}(\overline{p}_1, \overline{p}_2, \overline{e}) + 1 \right\} & \textbf{if } p_1 \neq e \wedge p_2 \neq e \\ T_{\mathbf{HJI}}(p_1, p_2, e) = 0 & \textbf{if } p_1 = e \vee p_2 = e \end{cases}$$



Modelling power pills via Hybrid Systems

- **Hybrid system**
  - **If Pacman gets a power pill, the ghosts switch their dynamics, running at maximal distance from the pill (using $T_{\mathbf{HJB}}$)**
  - **When the ghosts reach their target, they switch back to the pursuit dynamics**
- **Hierarchical Varifold**
  - **create a copy of the state space for each switch**
  - **order and connect different layers at switching points**



$1st$ **layer: pursuit-evasion dynamics**

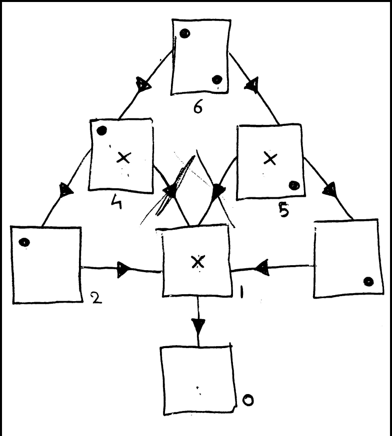
$2nd$ **layer: optimal drift for ghosts**

$3rd$ **layer: pursuit-evasion dynamics**

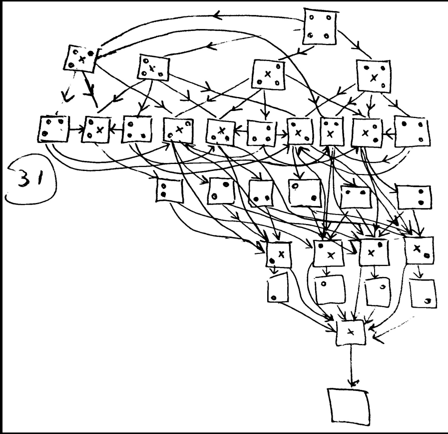
Compute solution backward & bottom-top, use value function restricted at switching points as boundary data for the next layer

Modelling power pills via Hybrid Systems

2 power pills ⟹ 7 layers

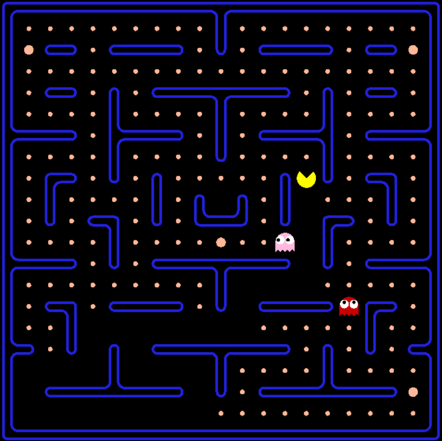


4 power pills ⟹ 31 layers



The full game... an open problem!

- Take all the pills (Salesman Problem)
- Time dependent problem?
- How to combine with pursuit-evasion game?



Pacman^{HJ} at European Researchers' Night

Department of Mathematics and Physics University Roma Tre
29 September 2017



PACMAN^{HJ}

A classical arcade videogame
powered by Hamilton-Jacobi equations

GAME OVER

Thank you!