From open-loop representations to closed-loop feedback implementations in differential games: A numerical case study

Philipp Braun

School of Engineering, The Australian National University

In Collaboration with:

T. L. Molloy	Monash University, Melbourne, Australia
G. Barkai	University of Lorraine, France
I. Shames	University of Melbourne, Australia

Funding

United States Air Force Office of Scientific Research under Grant No. FA2386-24-1-4014.

(Non-cooperative) Differential (two player) games

Dynamical systems:

- Player 1: $\dot{\xi}_1 = f_1(\xi_1, u_1)$
- Player 2: $\dot{\xi}_2 = f_2(\xi_2, u_2)$

with inputs u_1, u_2 and states ξ_1, ξ_2.

(Non-cooperative) Differential (two player) games

Dynamical systems:

- Player 1: $\dot{\xi}_1 = f_1(\xi_1, u_1)$
- Player 2: $\dot{\xi}_2 = f_2(\xi_2, u_2)$

with inputs u_1, u_2 and states ξ_1, ξ_2.

More generally, coupled dynamics:

$$\dot{\xi} = f(\xi, u_1, u_2)$$

with overall state ξ and inputs u_1, u_2 (corresponding to players 1 and 2).

(Non-cooperative) Differential (two player) games

Dynamical systems:

- Player 1: $\dot{\xi}_1 = f_1(\xi_1, u_1)$
- Player 2: $\dot{\xi}_2 = f_2(\xi_2, u_2)$

with inputs u_1, u_2 and states ξ_1, ξ_2.

More generally, coupled dynamics:

$$\dot{\xi} = f(\xi, u_1, u_2)$$

with overall state ξ and inputs u_1, u_2 (corresponding to players 1 and 2).

Differential game

$$\begin{aligned} V(\xi_0) = \min_{u_1(\cdot)} \max_{u_2(\cdot)} & \int_0^T L(\xi(t), u_1(t), u_2(t))\, dt + G(x(T)) \\ \text{s.t.} \quad & \xi(0) = \xi_0 && \leftarrow \text{Initial condition} \\ & \dot{\xi}(t) = f(\xi(t), u_1(t), u_2(t)) && \leftarrow \text{Dynamics} \\ & (u_1(t), u_2(t)) \in \mathcal{U}_1 \times \mathcal{U}_2 && \leftarrow \text{Input constraints} \\ & \xi(0) \in \mathcal{X}_0 && \leftarrow \text{Domain of interest/Game set} \\ & (\xi(T), \dot{\xi}(T)) \in \mathcal{X}_T && \leftarrow \text{Terminating condition} \end{aligned}$$

Additionally:

- $V(\cdot)$: Optimal value fcn
- $L(\cdot)$: Running costs
- $G(\cdot)$: Terminal costs

(Non-cooperative) Differential (two player) games

Dynamical systems:

- Player 1: $\dot{\xi}_1 = f_1(\xi_1, u_1)$
- Player 2: $\dot{\xi}_2 = f_2(\xi_2, u_2)$

with inputs u_1, u_2 and states ξ_1, ξ_2.

More generally, coupled dynamics:

$$\dot{\xi} = f(\xi, u_1, u_2)$$

with overall state ξ and inputs u_1, u_2 (corresponding to players 1 and 2).

Differential game

$$
\begin{aligned}
V(\xi_0) = \min_{u_1(\cdot)} \max_{u_2(\cdot)} & \int_0^T L(\xi(t), u_1(t), u_2(t))\, dt + G(x(T)) \\
\text{s.t.} \quad & \xi(0) = \xi_0 && \leftarrow \text{Initial condition} \\
& \dot{\xi}(t) = f(\xi(t), u_1(t), u_2(t)) && \leftarrow \text{Dynamics} \\
& (u_1(t), u_2(t)) \in \mathcal{U}_1 \times \mathcal{U}_2 && \leftarrow \text{Input constraints} \\
& \xi(0) \in \mathcal{X}_0 && \leftarrow \text{Domain of interest/Game set} \\
& (\xi(T), \dot{\xi}(T)) \in \mathcal{X}_T && \leftarrow \text{Terminating condition}
\end{aligned}
$$

Additionally:

- $V(\cdot)$: Optimal value fcn
- $L(\cdot)$: Running costs
- $G(\cdot)$: Terminal costs

We are interested in

- opt. input $u_1^*(t)$, $u_2^*(t)$
- opt. feedback laws $u_1^*(\xi(t))$, $u_2^*(\xi(t))$

Examples of (classical) Differential (two player) games

Lion and Man (Pursuit-evasion game):

- Fast agile agent (Lion) attempts to catch a slower agile agent (Man). Lion minimizes capture time; Man maximizes the capture time.

Agile agent:

$$\dot{x}_1(t) = v_1 \sin u_1(t)$$
$$\dot{y}_1(t) = v_1 \cos u_1(t).$$

- constant speed: $v_1 \geq 0$
- heading: $u_1 \in (-\pi, \pi]$ (input)

Examples of (classical) Differential (two player) games

Lion and Man (Pursuit-evasion game):

- Fast agile agent (Lion) attempts to catch a slower agile agent (Man). Lion minimizes capture time; Man maximizes the capture time.

Homicidal chauffeur:

- Runner (slow, agile agent), against chauffeur of a motor vehicle (fast, less maneuverable). Chauffeur attempts to run the runner down (in minimal time). Runner attempts to avoid the vehicle (as long as possible)

Agile agent:

$$\dot{x}_1(t) = v_1 \sin u_1(t)$$
$$\dot{y}_1(t) = v_1 \cos u_1(t).$$

- constant speed: $v_1 \geq 0$
- heading: $u_1 \in (-\pi, \pi]$ (input)

Less maneuverable agent (Dubins vehicle):

$$\dot{x}_2(t) = v_2 \sin \theta_2(t)$$
$$\dot{y}_2(t) = v_2 \cos \theta_2(t)$$
$$\dot{\theta}_2(t) = \omega_2 u_2(t).$$

- constant speed: v_2
- maximum turn rate: ω_2
- norm. turn rate: $u_2(t) \in [-1, 1]$ (inp.)

Examples of (classical) Differential (two player) games

Lion and Man (Pursuit-evasion game):

- Fast agile agent (Lion) attempts to catch a slower agile agent (Man). Lion minimizes capture time; Man maximizes the capture time.

Homicidal chauffeur:

- Runner (slow, agile agent), against chauffeur of a motor vehicle (fast, less maneuverable). Chauffeur attempts to run the runner down (in minimal time). Runner attempts to avoid the vehicle (as long as possible)

Suicidal pedestrian:

- Reversed "homicidal chauffeur". The roles of the runner and the chauffeur are switched.

Agile agent:

$$\begin{aligned}\dot{x}_1(t) &= v_1 \sin u_1(t)\\ \dot{y}_1(t) &= v_1 \cos u_1(t).\end{aligned}$$

- constant speed: $v_1 \geq 0$
- heading: $u_1 \in (-\pi, \pi]$ (input)

Less maneuverable agent (Dubins vehicle):

$$\begin{aligned}\dot{x}_2(t) &= v_2 \sin \theta_2(t)\\ \dot{y}_2(t) &= v_2 \cos \theta_2(t)\\ \dot{\theta}_2(t) &= \omega_2 u_2(t).\end{aligned}$$

- constant speed: v_2
- maximum turn rate: ω_2
- norm. turn rate: $u_2(t) \in [-1, 1]$ (inp.)

Examples of (classical) Differential (two player) games

Lion and Man (Pursuit-evasion game):

- Fast agile agent (Lion) attempts to catch a slower agile agent (Man). Lion minimizes capture time; Man maximizes the capture time.

Homicidal chauffeur:

- Runner (slow, agile agent), against chauffeur of a motor vehicle (fast, less maneuverable). Chauffeur attempts to run the runner down (in minimal time). Runner attempts to avoid the vehicle (as long as possible)

Suicidal pedestrian:

- Reversed "homicidal chauffeur". The roles of the runner and the chauffeur are switched.

Pursuit-evasion & surveillance-evasion games:

- Pursuing agent attempts to catch evading agent
- Evading agent attempts to escape surveillance radius of pursing agent

Agile agent:

$$\dot{x}_1(t) = v_1 \sin u_1(t)$$
$$\dot{y}_1(t) = v_1 \cos u_1(t).$$

- constant speed: $v_1 \geq 0$
- heading: $u_1 \in (-\pi, \pi]$ (input)

Less maneuverable agent (Dubins vehicle):

$$\dot{x}_2(t) = v_2 \sin \theta_2(t)$$
$$\dot{y}_2(t) = v_2 \cos \theta_2(t)$$
$$\dot{\theta}_2(t) = \omega_2 u_2(t).$$

- constant speed: v_2
- maximum turn rate: ω_2
- norm. turn rate: $u_2(t) \in [-1, 1]$ (inp.)

Examples of (classical) Differential (two player) games

Prying pedestrian: (A surveillance-evasion game)

- Slow, agile agent attempts to maximize surveillance of fast less maneuverable agent
- Fast, less maneuverable agent attempts to minimize surveillance of slow agile agent

[1] **P. Braun, T. L. Molloy and I. Shames;** *Prying Pedestrian Surveillance-Evasion: Minimum-Time Evasion from an Agile Pursuer*, Journal of Guidance, Control, and Dynamics 48 (9), 2090-2104, 2025

Agile agent:

$$\dot{x}_1(t) = v_1 \sin u_1(t)$$
$$\dot{y}_1(t) = v_1 \cos u_1(t).$$

- constant speed: $v_1 \geq 0$
- heading: $u_1 \in (-\pi, \pi]$ (input)

Less maneuverable agent (Dubins vehicle):

$$\dot{x}_2(t) = v_2 \sin \theta_2(t)$$
$$\dot{y}_2(t) = v_2 \cos \theta_2(t)$$
$$\dot{\theta}_2(t) = \omega_2 u_2(t).$$

- constant speed: v_2
- maximum turn rate: ω_2
- norm. turn rate: $u_2(t) \in [-1, 1]$ (inp.)

Prying Pedestrian Differential Game

Fast Evader ($u_e \in [-1, 1]$)

$$\dot{\xi}_e = \begin{bmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{\theta}_e \end{bmatrix} = \begin{bmatrix} v_e \sin \theta_e(t) \\ v_e \cos \theta_e(t) \\ \omega_e u_e(t). \end{bmatrix}$$

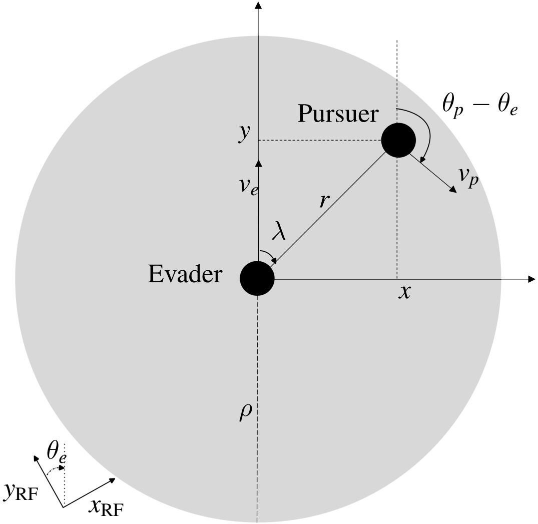
Agile Pursuer ($\theta_p \in \mathbb{R}$)

$$\dot{\xi}_p = \begin{bmatrix} \dot{x}_p(t) \\ \dot{y}_p(t) \end{bmatrix} = \begin{bmatrix} v_p \sin \theta_p(t) \\ v_p \cos \theta_p(t) \end{bmatrix}$$

Evader Centric Coordinates

$$\dot{\xi}(t) = f(\xi(t), u_e(t), u_p(t)), \quad \xi = \begin{bmatrix} x \\ y \end{bmatrix}$$

$$f(\xi, u_e, u_p) = \begin{bmatrix} -\omega_e y u_e + v_p \sin u_p \\ \omega_e x u_e - v_e + v_p \cos u_p \end{bmatrix}.$$



Prying Pedestrian Differential Game

Fast Evader ($u_e \in [-1, 1]$)

$$\dot{\xi}_e = \begin{bmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{\theta}_e \end{bmatrix} = \begin{bmatrix} v_e \sin\theta_e(t) \\ v_e \cos\theta_e(t) \\ \omega_e u_e(t). \end{bmatrix}$$

Agile Pursuer ($\theta_p \in \mathbb{R}$)

$$\dot{\xi}_p = \begin{bmatrix} \dot{x}_p(t) \\ \dot{y}_p(t) \end{bmatrix} = \begin{bmatrix} v_p \sin\theta_p(t) \\ v_p \cos\theta_p(t) \end{bmatrix}$$

Evader Centric Coordinates

$$\dot{\xi}(t) = f(\xi(t), u_e(t), u_p(t)), \quad \xi = \begin{bmatrix} x \\ y \end{bmatrix}$$

$$f(\xi, u_e, u_p) = \begin{bmatrix} -\omega_e y u_e + v_p \sin u_p \\ \omega_e x u_e - v_e + v_p \cos u_p \end{bmatrix}.$$

Prying pedestrian Surveillance Evasion Game

$$V(\xi_0) = \min_{u_e:[0,T]\to[-1,1]} \max_{u_p:[0,T]\to\mathbb{R}} \int_0^T 1 \, dt$$

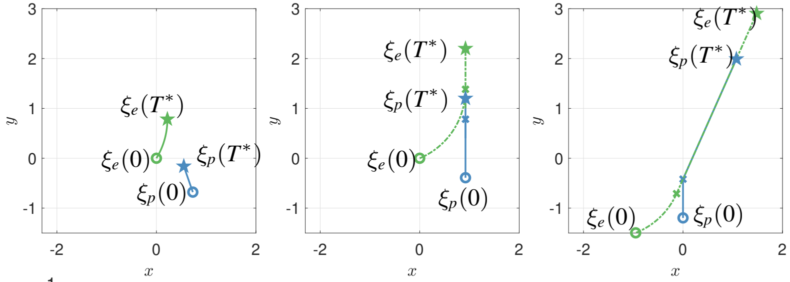
$$\text{s. t.} \quad \dot{\xi}(t) = f(\xi(t), u_e(t), u_p(t)), \quad \xi(0) = \xi_0$$

$$|\xi(0)|_2 \le \rho, \quad |\xi(T)|_2 = \rho, \quad \frac{d}{dt}|\xi(T)|_2^2 > 0$$

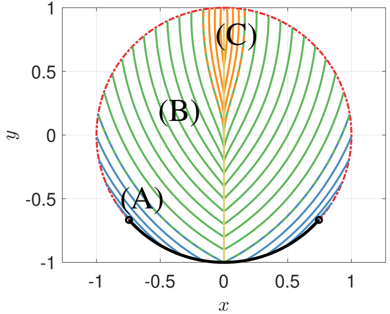
A fast Evader (E) escapes the surveillance of an agile Pursuer (P) in minimum time

Optimal Value function $V(\xi_0)$ represents optimal time to end the game.

Solution of the game of degree



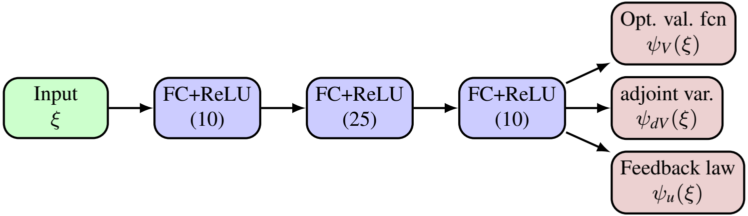
Solution Concept:
Backward reachability analysis via HJI equation



Implicit feedback characterization (and adjoint variables $\nabla V(\cdot)$):

$$
\begin{aligned}
u_e^*(\xi) &= \arg\min_{u_e \in [-1,1]} \left[(\nabla_y V(\xi) x - \nabla_x V(\xi) y)\omega_e u_e\right] \\
&= \operatorname{sign}(\nabla V_x(\xi) y - \nabla V_y(\xi) x) \\
u_p^*(\xi) &= \arg\max_{u_p(\xi) \in \mathbb{R}} \left[(\nabla V_x(\xi) \sin u_p + \nabla V_y(\xi) \cos u_p) v_p\right] \\
&= \arctan_2\left(\nabla V_x(\xi), \nabla V_y(\xi)\right)
\end{aligned}
$$

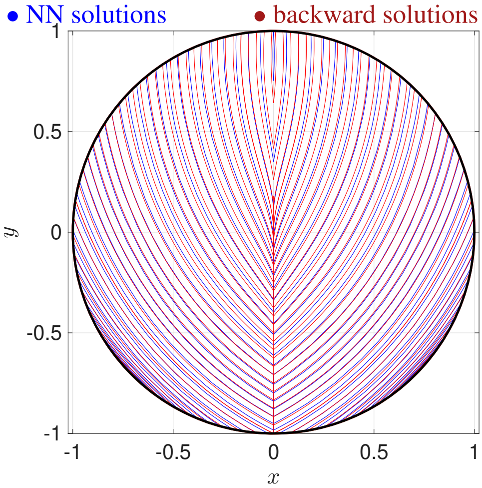
From open-loop solutions to feedback laws (using neural networks)



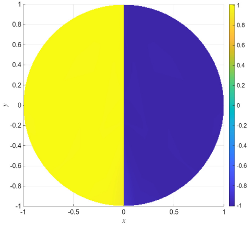
Loss function: $\quad (\kappa_{10}(x) = 10|x|_1 + |x|_2^2)$

$$
\begin{aligned}
L(\xi) = {} & \kappa_{10}(\psi_V(\xi) - V(\xi)) + \kappa_{10}(\psi_{dV}(\xi) - \nabla V(\xi)) && \leftarrow \text{Penalize (derivative of) opt. val. fcn.} \\
& + \kappa_{10}\left(\psi_u(\xi) - \begin{bmatrix} u_e^*(\xi) \\ u_p^*(\xi) \end{bmatrix}\right) && \leftarrow \text{Penalize input} \\
& + \kappa_{10}\left(\nabla\psi_V(\xi) - \psi_{dV}(\xi)\right) && \leftarrow \text{Penalize NN output mismatch} \\
& + \kappa_{10}\left(\begin{bmatrix} \operatorname{sign}(\psi_{dV,1}(\xi)\xi_2 - \psi_{dV,2}\xi_1) - \psi_{u,1}(\xi) \\ \arctan_2(\psi_{dV,1}(\xi), \psi_{dV,2}(\xi)) - \psi_{u,2}(\xi) \end{bmatrix}\right) && \leftarrow \text{Penalize implicit feedback char.}
\end{aligned}
$$

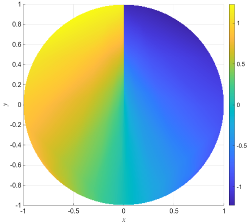
Solutions and Feedback laws



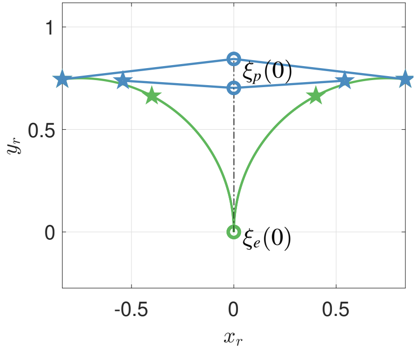
Evader feedback: $\psi_{u,1}(\xi)$

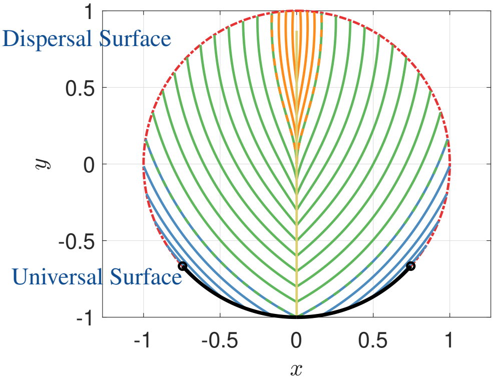


Pursuer feedback: $\psi_{u,2}(\xi)$



Robustness analysis of feedback laws





"Realistic dynamics":

$$\dot{\xi} = f(\xi + n, u_e + d_e, u_p + d_p) + m, \qquad \text{with noise } n, \text{ disturbances } d_e, d_p, \text{ uncertainty } m.$$

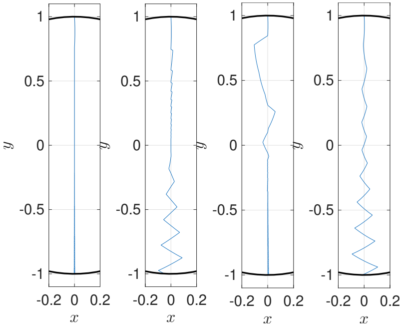
$\rightsquigarrow$ In which direction should pursuer go?

Sample-and-hold implementation

Sample-and-hold feedback implementation

$$\dot{\xi}(t) = f(\xi, \bar{\psi}_{u,1}(t), \bar{\psi}_{u,2}(t)), \qquad \xi(0) = \xi_0$$

- $\bar{\psi}_{u,1}(t) = \psi_{u,1}(\xi(\delta_e k)) \quad \forall t \in [\delta_e k, \delta_e(k+1))$
- $\bar{\psi}_{u,2}(t) = \psi_{u,2}(\xi(\delta_p k)) \quad \forall t \in [\delta_p k, \delta_p(k+1))$



Sampling times:

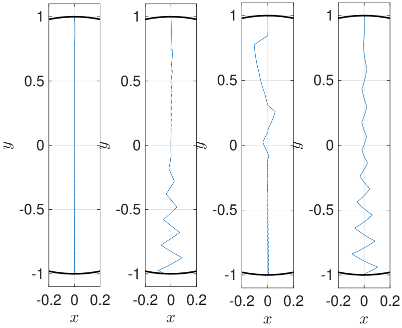
$$\begin{bmatrix} \delta_e \\ \delta_p \end{bmatrix} \in \left\{ \begin{bmatrix} 0.01 \\ 0.01 \end{bmatrix}, \begin{bmatrix} 0.2 \\ 0.01 \end{bmatrix}, \begin{bmatrix} 0.01 \\ 0.2 \end{bmatrix}, \begin{bmatrix} 0.2 \\ 0.2 \end{bmatrix} \right\}$$

Sample-and-hold implementation

Sample-and-hold feedback implementation

$$\dot{\xi}(t) = f(\xi, \bar{\psi}_{u,1}(t), \bar{\psi}_{u,2}(t)), \qquad \xi(0) = \xi_0$$

- $\bar{\psi}_{u,1}(t) = \psi_{u,1}(\xi(\delta_e k)) \quad \forall t \in [\delta_e k, \delta_e (k+1))$
- $\bar{\psi}_{u,2}(t) = \psi_{u,2}(\xi(\delta_p k)) \quad \forall t \in [\delta_p k, \delta_p (k+1))$



Opt. val. fcn interpretation:

- Benchmark solution: $V_{0.01}^{0.01}(\xi_0) = 3.444$
- Evader is not disadvantaged: $V_{0.01}^{0.2}(\xi_0) = 3.441$
- Evader benefits: $V_{0.2}^{0.01}(\xi_0) = 3.228$
- Evader benefits: $V_{0.2}^{0.2}(\xi_0) = 3.309$

Sampling times:

$$\begin{bmatrix} \delta_e \\ \delta_p \end{bmatrix} \in \left\{ \begin{bmatrix} 0.01 \\ 0.01 \end{bmatrix}, \begin{bmatrix} 0.2 \\ 0.01 \end{bmatrix}, \begin{bmatrix} 0.01 \\ 0.2 \end{bmatrix}, \begin{bmatrix} 0.2 \\ 0.2 \end{bmatrix} \right\}$$

Rate of loss/gain* (through sample-and-hold-feedback)

One-step Pursuer (P) loss:

$$V_{\min}^{\delta}(\xi_0) = \min_{u_e \in [-1,1],\; u_p^*(\xi_0)} \int_0^{\delta} 1 \, dt + V(\xi(\delta))$$

$V_{\min}^{\delta}(\xi_0)$: approximates how much P can lose (i.e., early termination) if E knows that P uses $u_p(t) = u_p^*(\xi_0)$ for $t \in [0, \delta]$

*D. Milutinović, D. W. Casbeer, A. Von Moll, M. Pachter and E. Garcia, Eloy; *Rate of Loss Characterization That Resolves the Dilemma of the Wall Pursuit Game Solution*, IEEE TAC, 68(1), 242–256, 2023

Rate of loss/gain* (through sample-and-hold-feedback)

One-step Pursuer (P) loss:

$$V^{\delta}_{\min}(\xi_0) = \min_{u_e \in [-1,1],\ u_p^*(\xi_0)} \int_0^{\delta} 1\, dt + V(\xi(\delta))$$

$V^{\delta}_{\min}(\xi_0)$: approximates how much P can lose (i.e., early termination) if E knows that P uses $u_p(t) = u_p^*(\xi_0)$ for $t \in [0, \delta]$

One-step Evader (E) gain:

$$V^{\delta}_{\max}(\xi_0) = \max_{u_e^*(\xi_0),\ u_p \in \mathbb{R}} \int_0^{\delta} 1\, dt + V(\xi(\delta))$$

$V^{\delta}_{\max}(\xi_0)$: approximates how much E can gain (i.e., game prolongation) if P knows that E uses $u_e(t) = u_e^*(\xi_0)$ for $t \in [0, \delta]$

*D. Milutinović, D. W. Casbeer, A. Von Moll, M. Pachter and E. Garcia, Eloy; *Rate of Loss Characterization That Resolves the Dilemma of the Wall Pursuit Game Solution*, IEEE TAC, 68(1), 242–256, 2023

Rate of loss/gain* (through sample-and-hold-feedback)

One-step Pursuer (P) loss:

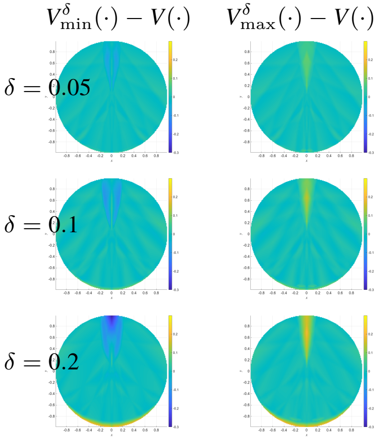
$$V^{\delta}_{\min}(\xi_0) = \min_{u_e \in [-1,1],\ u_p^*(\xi_0)} \int_0^{\delta} 1\, dt + V(\xi(\delta))$$

$V^{\delta}_{\min}(\xi_0)$: approximates how much P can lose (i.e., early termination) if E knows that P uses $u_p(t) = u_p^*(\xi_0)$ for $t \in [0, \delta]$

One-step Evader (E) gain:

$$V^{\delta}_{\max}(\xi_0) = \max_{u_e^*(\xi_0),\ u_p \in \mathbb{R}} \int_0^{\delta} 1\, dt + V(\xi(\delta))$$

$V^{\delta}_{\max}(\xi_0)$: approximates how much E can gain (i.e., game prolongation) if P knows that E uses $u_e(t) = u_e^*(\xi_0)$ for $t \in [0, \delta]$



*D. Milutinović, D. W. Casbeer, A. Von Moll, M. Pachter and E. Garcia, Eloy; *Rate of Loss Characterization That Resolves the Dilemma of the Wall Pursuit Game Solution*, IEEE TAC, 68(1), 242–256, 2023

Conclusions & Outlook

Building on

- open-loop solutions of the Prying Pedestrian Surveillance evasion game
- we have used NNs to obtain feedback control laws

Using NN based feedback controllers

- we have numerically analysed the rate of loss/gain of (asymmetric) sample-and-hold feedback implementations

Future work will focus on

- an analytical analysis of the rate of loss/gain
- the rate of loss/gain in other differential games

Our NN is based on a fixed set of parameters.

- Can parameters (speed, turn rate, surveillance radius) be considered as additional input?
- Can we 'learn' the opponents parameters online?