Algoritmi di pianificazione dinamica basati su Velocity Obstacle - VO

algorithms for robot collision avoidance

Davide Zorzi

Universitá degli Studi di Verona

davide.zorzi@studenti.univr.it

15 novembre 2016

Overview

Definizione del problema

Ogni agente é indipendente uno dall'altro senza coordinate centrali e senza comunicare con gli altri agenti. Consideriamo che ogni agente (ostacolo) sia a *disc-shape* nell'ambiente.

Requisiti base

Ogni algoritmo prevede la conoscenza di alcune informazioni rese pubbliche a ogni iterazione del ciclo di *sensing-acting*.
Per l'agente A:

- $\boldsymbol{r_A}$ raggio dell'agente
- $\boldsymbol{p_A}$ posizione corrente
- $\boldsymbol{v_A}$ velocitá corrente

Informazioni non condivise nella scena:

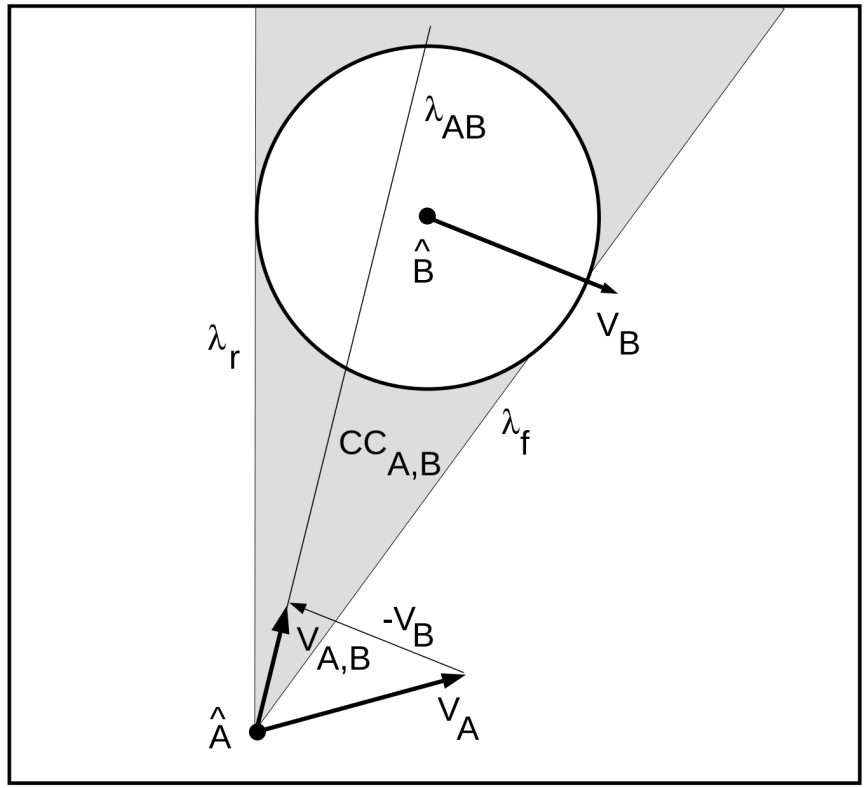
- $\boldsymbol{p^{goal}{}_A}$ posizione di arrivo
- $\boldsymbol{v^{pref}{}_A}$ velocitá preferita

Collision Cone

Definiamo *Collision Cone*, $CC_{A,B}$, l'insieme delle *colliding relative velocities* tra ***A'*** e ***B'***:

$$CC_{A,B} = \{\boldsymbol{v}_{A,B} | \lambda_{A,B} \cap \boldsymbol{B'} \neq \oslash\} \tag{1}$$

dove $\boldsymbol{v}_{\mathbf{A,B}}$ rappresenta la velocitá relativa di ***A'*** rispetto ***B'***, $\boldsymbol{v}_{\mathbf{A,B}} = \boldsymbol{v}_{\mathbf{A}}$ - $\boldsymbol{v}_{\mathbf{B}}$, e $\lambda_{\mathrm{A,B}}$ é la linea di $\boldsymbol{v}_{\mathbf{A,B}}$. Questo cono é la parte di piano delimitata da tue tangenti, λ_{f} e λ_{r}, con apice in ***A'***.

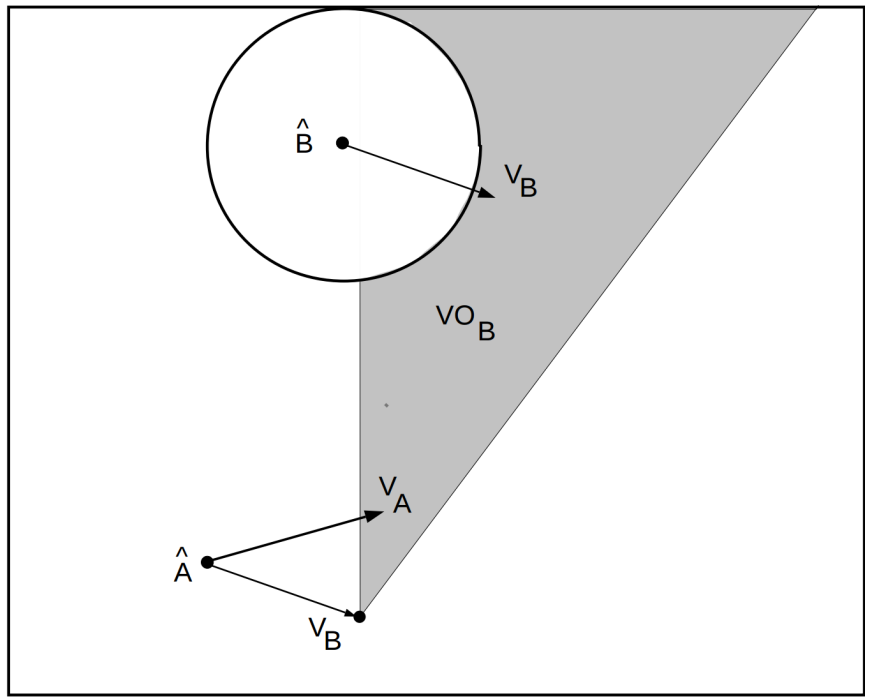


Velocity Obstacle

Velocity Obstacle VO é definito:

$$VO = CC_{A,B} \oplus \boldsymbol{v}_B \qquad (2)$$

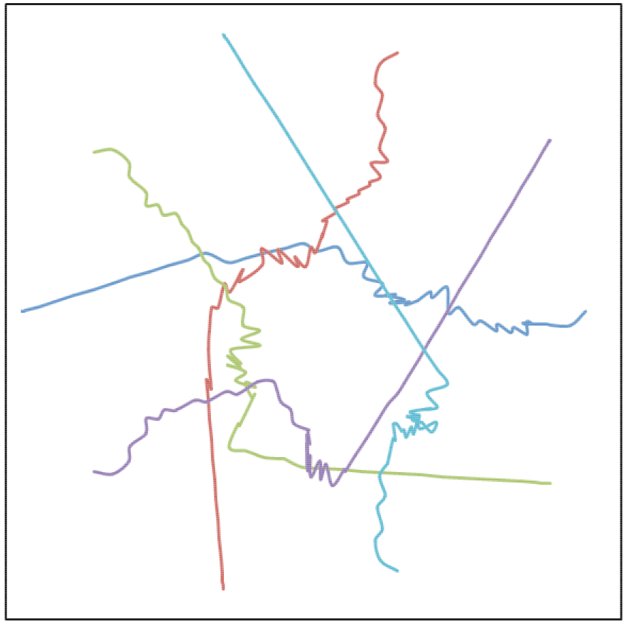
dove $\oplus$ é l' operatore
somma vettoriale di Minkowski.



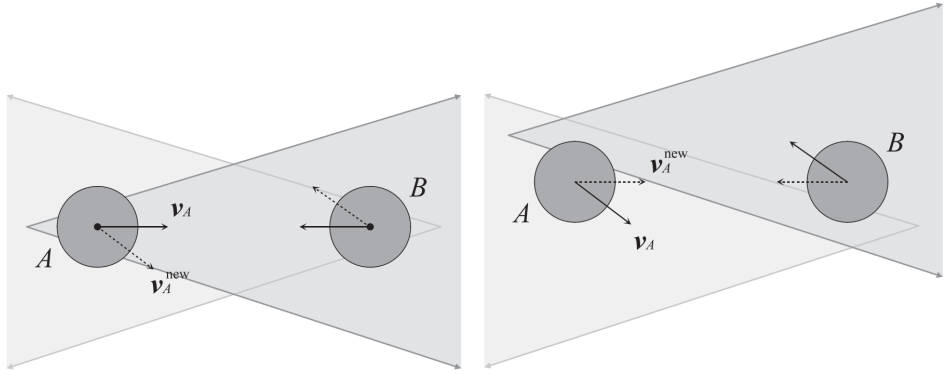
Oscillations

Oscillations

- Si possono verificare oscillazioni dal momento che gli agenti sono entitá decisionali
- Ogni selezione dalla velocitá é fuori da ogni VO degli altri agenti, ma le vecchie velocitá sono ancora valide con il rispetto alle nuove VO.
- Gli agenti oscilleranno tra queste velocitá



(a) Traccia VO



(b) Oscilla tra v_A e v^{new}_A

Reciprocal Velocity Obstacle

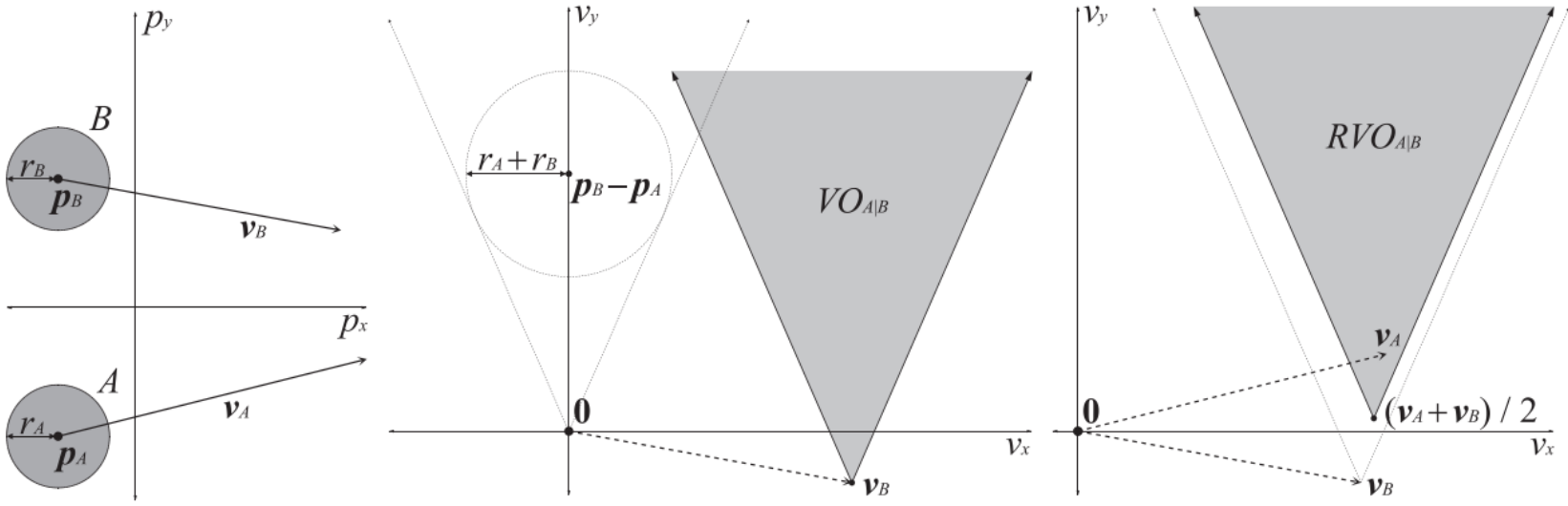
La nuova velocitá calcolata con questo algoritmo sia la media (*average*) della velocitá corrente e di una velocitá che sia al di fuori degli Velocity Obstacle degli altri agenti.

Definizione

$$RVO_{A,B}(\boldsymbol{v}_B, \boldsymbol{v}_A) = \{\boldsymbol{v'}_A | 2\boldsymbol{v'}_A - \boldsymbol{v}_A \in VO_{A,B}(\boldsymbol{v}_B)\} \tag{3}$$

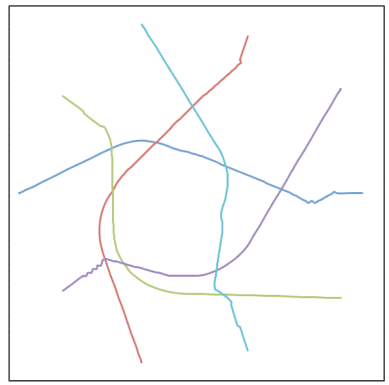
dove $\mathrm{RVO_{A,B}}(\boldsymbol{v_B}, \boldsymbol{v_A})$ dell'agente B su l'agente A contiene tutte le velocitá per A che saranno la media della velocitá corrente $\boldsymbol{v_A}$ e una velocitá all'interno di $\mathrm{VO_{A,B}}(\boldsymbol{v_B})$ dell'agente B. Puó essere interpretato geometricamente come Velocity Obstacle, $\mathrm{VO_{A,B}}(\boldsymbol{v_B})$, traslato di $\frac{\boldsymbol{v}_A + \boldsymbol{v}_B}{2}$ dall'apice.

RVO



- L'insieme delle velocitá di VO é la media con la velocitá corrente.
- Un agente prende la metá della responsabilitá per evitare la collisione e l'altro agente prenderá altra metá.
- Ogni agente per non avere collisioni seleziona la velocitá al di fuori dei RVO degli altri.

Reciprocal Dances

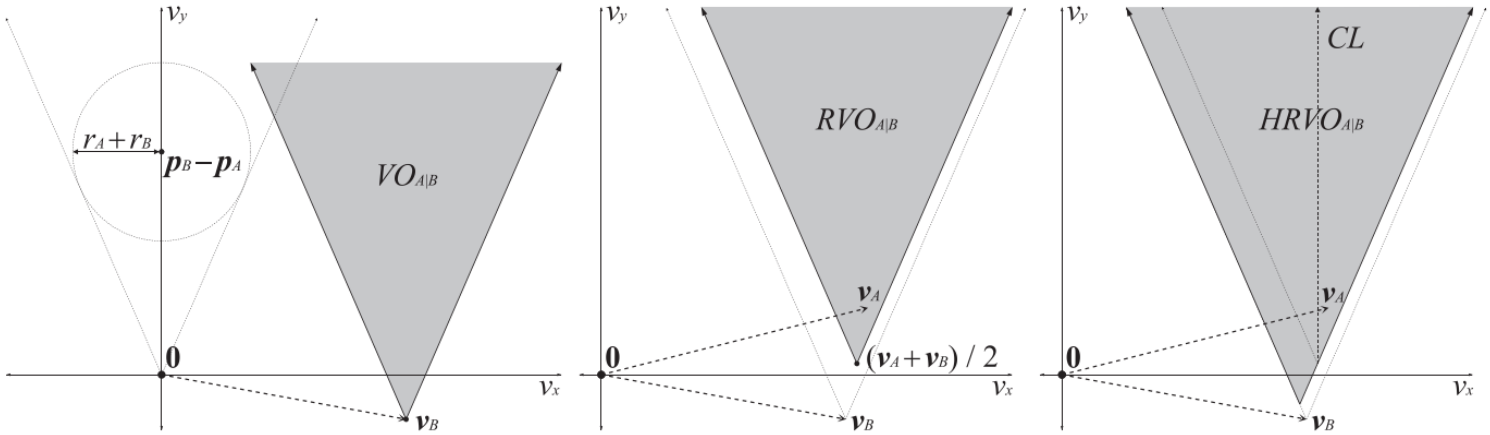


(c) Traccia RVO

Reciprocal Dances

- Reciprocal dances si possono verificare oscillazioni quando gli agenti non riescono a selezionare una velicitá libera vicina alla velocitá corrente.
- Puó essere dovuta a scegliere la velocitá verso il punto di arrivo o a causa della presenza di un terzo agente.

Hybrid Reciprocal Velocity Obstacle



HRVO

- RVO é allargato sul lato dove non vogliamo che l'agente passi.
- Sostituendo il bordo del RVO su quel lato con il bordo del VO.

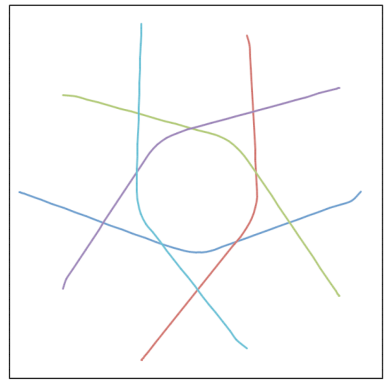
New Velocity

La computazione della nuova velocitá chiamata $\boldsymbol{v}^{\mathbf{new}}{}_{\mathbf{Ai}}$, al di fuori della combinazione dei HRVO, é la minima distanza tra la velocitá corrente e la velocitá preferita:

$$v_{Ai}^{new} = argmin_{v \notin HRVO_{Ai}} norm(v - v_{Ai}^{pref}). \tag{4}$$

Per computare tale velocitá viene utilizzato un algoritmo chiamato ClearPath, combinando tutti gli *HRVO* come intersezioni di segmenti. Le possibili velocitá ammissibili saranno tutte le intersezioni dei coni sul bordo di *HRVO*. Aggiungendo la proiezione della velocitá preferita, la nuova velocitá sará selezionata tra la minima distanza tra le velocitá ammissibili (intersezioni dei coni) e la velocitá preferita del robot $\boldsymbol{v}^{\mathbf{pref}}{}_{\mathbf{Ai}}$.
Se non si riesce a trovare una soluzione la procedura viene ripetuta.

No oscillations or Reciprocal dances

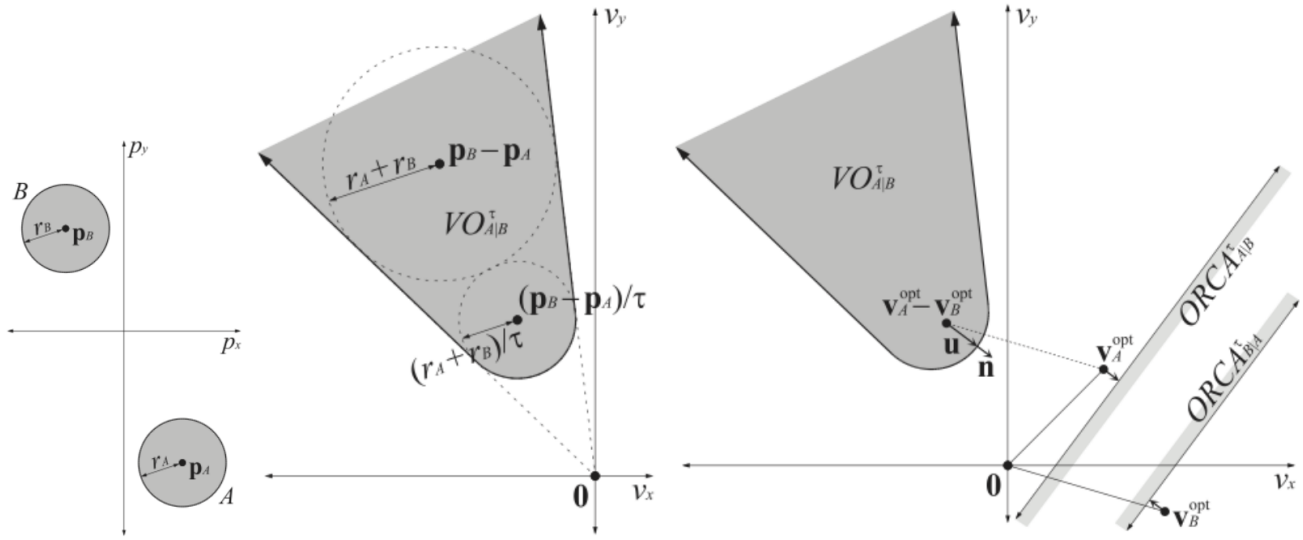


(d) Traccia HRVO

No oscillations or Reciprocal dances

- Se A tenta di passare sul lato sbagliato di B, deve dare la piena prioritá a B, come con VO.
- Se A sceglie il lato corretto, puó assumere la completa cooperazione con B e mantenere la paritá di prioritá, come con RVO.
- Riduce la Reciprocal Dances delle oscillazioni senza eccessivi movimenti vincolati.

Optimal reciprocal Collision Avoidance



Per costruire geometricamente $ORCA^{\tau}{}_{A,B}$ e $ORCA^{\tau}{}_{B,A}$ assumendo che A e B adottino rispettivamente $\boldsymbol{v}^{\mathbf{opt}}{}_{\mathbf{A}}$ e $\boldsymbol{v}^{\mathbf{opt}}{}_{\mathbf{B}}$, assumiamo che A e B siano in collisione se $\boldsymbol{v}^{\mathbf{opt}}{}_{\mathbf{A}}$ - $\boldsymbol{v}^{\mathbf{opt}}{}_{\mathbf{B}} \in VO^{\tau}{}_{A,B}$.

$$u = (argmin_{v \in VO^{\tau}_{A,B}} ||v - (v_A^{opt} - v_B^{opt})||) - (v_A^{opt} - v_B^{opt}), \qquad (5)$$

$\boldsymbol{u}$ é il piú piccolo cambiamento richiesto per le velocitá relative di A e B, per accorgersi della collisione al tempo τ.

Semipiano ORCA

Per spartirsi la responsabilitá della collisone, il robot A adatta la sua velocitá per almeno $\frac{1}{2}$ $\boldsymbol{u}$ assumendo che B si prenda cura dell'altra parte. L'insieme delle velocitá permesse $ORCA^{\tau}{}_{\mathrm{A,B}}$ per A é un semipiano definito come segue:

$$ORCA^{\tau}_{A,B} = \{\boldsymbol{v} | (\boldsymbol{v} - (\boldsymbol{v}^{opt}_{A} + \frac{1}{2}\boldsymbol{u})) \geq 0\}. \tag{6}$$

L'insieme $ORCA^{\tau}{}_{\mathrm{B,A}}$ per B é definito simmetricamente. Le equazioni qui sopra riportate, si applicano anche se A e B non sono su una rotta di collisione quando adottano le loro velocitá di ottimizzazione, $\boldsymbol{v}^{\mathbf{opt}}{}_{\mathbf{A}}$- $\boldsymbol{v}^{\mathbf{opt}}{}_{\mathbf{A}}$ $\notin$ $VO^{\tau}{}_{\mathrm{A,B}}$. In questo caso, ogni robot prenderá metá della responsabilitá per rimanere in una traiettoria di *collision-free*.

Basic Approach

Ogni robot A esegue un ciclo continuo di *sensing* e *acting* a ogni time-step Δt. L'insieme delle velocitá permesse per A é l'intersezione di ogni semipiano:

$$ORCA_A^\tau = D(0, v_A^{max}) \cap \bigcap_{B \neq A} ORCA_{A,B}^\tau \tag{7}$$

Nel passo successivo, il robot seleziona la nuova velocitá $\boldsymbol{v}^{\mathbf{new}}{}_{\mathbf{A}}$ la quale sará la piú vicina possibile alla velocitá preferita $\boldsymbol{v}^{\mathbf{pref}}{}_{\mathbf{A}}$ tale che, questa velocitá, risiedi almeno all'interno dell'insieme delle velocitá permesse:

$$v_A^{new} = argmin_{v \in ORCA_A^\tau} ||v - v_A^{pref}||. \tag{8}$$

Finalmente, il robot riceverá la sua nuova posizione:

$$p_A^{new} = p_A + v_A^{new} \Delta t, \tag{9}$$

e il ciclo di *sensing-acting* verrá ripetuto.

Choosing the Optimization Velocity

Per la scelta della nuova velocitá in modo efficiente:

- $\boldsymbol{v}^{\mathbf{opt}}{}_{\mathbf{A}} = 0$ per ogni robot A. Per qualsiasi robot B, il punto 0 si trova sempre al di fuori del $VO^{\tau}{}_{\mathrm{A,B}}$. Quindi il semipiano, $ORCA^{\tau}{}_{\mathrm{A,B}}$, include sempre almeno la velocitá 0.
- $\boldsymbol{v}^{\mathbf{opt}}{}_{\mathbf{A}} = \boldsymbol{v}^{\mathbf{pref}}{}_{\mathbf{A}}$ per ogni robot A.
- $\boldsymbol{v}^{\mathbf{opt}}{}_{\mathbf{A}} = \boldsymbol{v}_{\mathbf{A}}$ per tutti i robot A. Impostare la velocitá corrente come velocitá ottimale, sarebbe il comportamento ideale.

Smooth Trajectories

Smooth Trajectories

- Dato un time-step abbastanza piccolo, la traiettoria generata dalla sequenza delle velocitá al centro dell'agente é continua nel velocity space.
- Dato un time-step abbastanza piccolo, la traiettoria generata dalla sequenza delle posizioni del centro dell'agente é collision-free con tutti gli altri percorsi rispetto al raggio dell'agente.

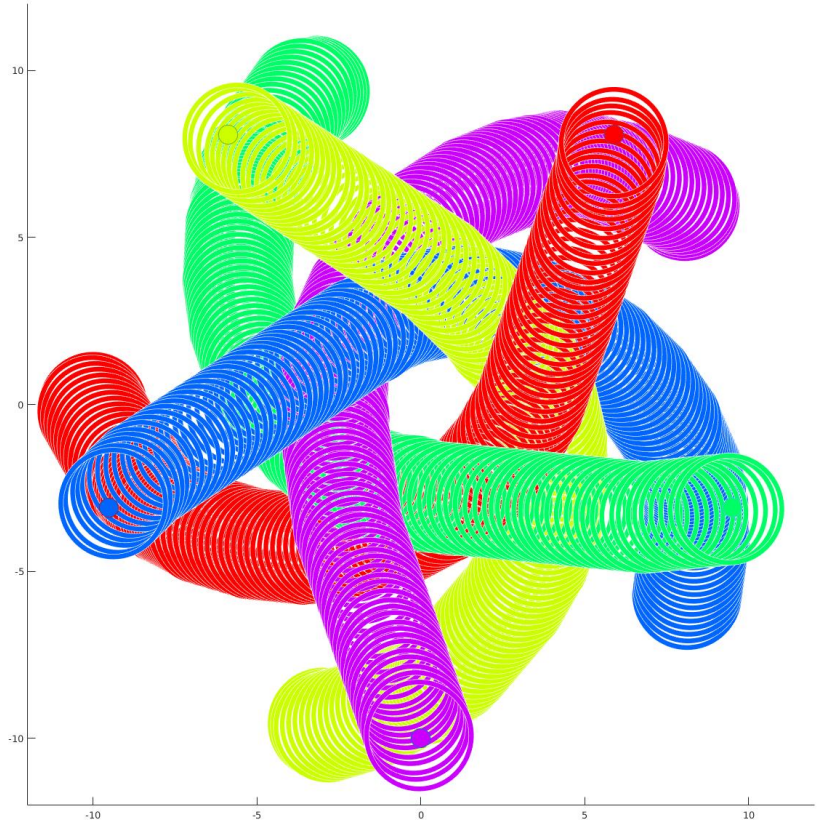
Detect Velocity Obstacle

Il calcolo della nuova velocitá é prevista dalla funzione *findVelocity* che prende in input l'agente e l'altro agente.

- Admin_Speeds: calcola tutte le possibili velocitá che puó assumere l'agente.
- cone_VO: calcola tutti i coni della scena per l'agente preso in osservazione
- inpolygon: gli viene passato il cono e le velocitá possibili
- ad_vel: é l'array risultante con tutte le velocitá candidate a essere la nuova velocitá
- min(ad_vel): prende la velocitá che differisce del minimo possibile dalla velocitá preferita
- aggiornamento della nuova velocitá, posizione e di conseguenza anche della velocitá preferita

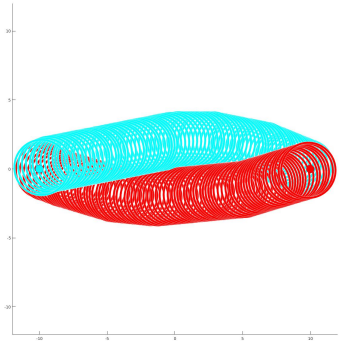
Result

Confrontando le traiettorie di VO (a), RVO (c) e HRVO (d) l'algortimo DVO presenta una traiettoria senza Oscillazioni e Reciprocal Dances.

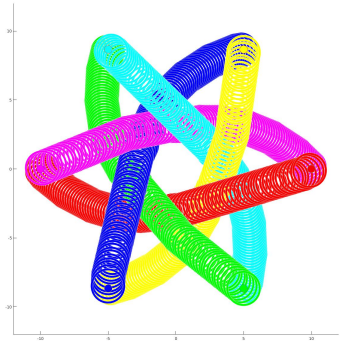


(e) Traccia DVO

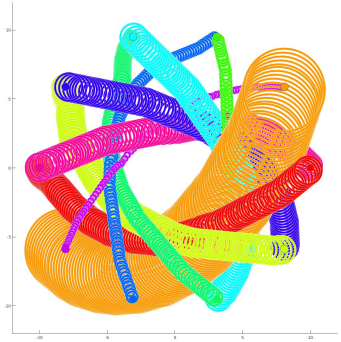
Atri scenari di DVO



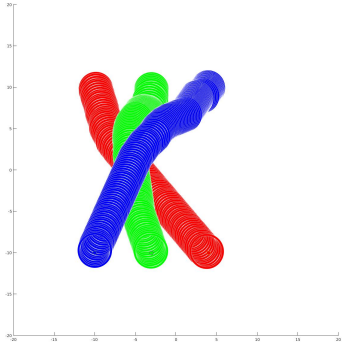
(f) Due agenti di raggio 2



(g) Sei agenti di raggio 1



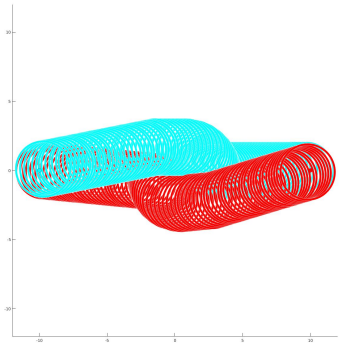
(h) Dodici agenti di raggio random



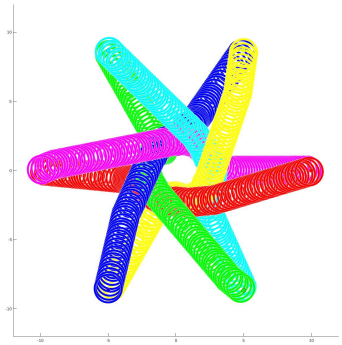
(i) Tre agenti di raggio 1

Figura: Stampa delle traiettorie della simulazione

Result con τ



(a) Due agenti di raggio 2



(b) Sei agenti di raggio 1

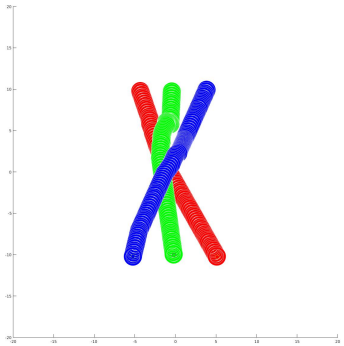


Figura: Stampa delle traiettorie della simulazione