Route planning problems and hybrid control

Roberto Ferretti

Department of Mathematics and Physics, Roma Tre University

ferretti@mat.uniroma3.it

ICODE Paris, 10.01.20

joint works with
S. Cacace (Roma Tre) and A. Festa (Torino)

Outline

State equations of a stochastic hybrid system (1)

- **State** of the system: $(X(t), Q(t)) \in \Omega \times \mathbb{I}$, with $\Omega \subseteq \mathbb{R}^d$, $\mathbb{I} = \{1, \ldots, Q_m\}$. The discrete variable $Q(t)$ (with initial value $q = Q(0)$) tells **which dynamics is active at time** t
- A measurable **control** $u(t)$ mapping $(0, +\infty)$ into a compact set U
- A stochastic term driven by the coefficient σ

State equation

Evolution for given initial values of X and Q:

$$\begin{cases} dX(t) = f(X(t), Q(t), u(t))dt + \sigma(X(t), Q(t))dW(t), \\ X(0) = x, \\ Q(0) = q. \end{cases}$$

State equations of a stochastic hybrid system (1)

- **State** of the system: $(X(t), Q(t)) \in \Omega \times \mathbb{I}$, with $\Omega \subseteq \mathbb{R}^d$, $\mathbb{I} = \{1, \dots, Q_m\}$. The discrete variable $Q(t)$ (with initial value $q = Q(0)$) tells **which dynamics is active at time** t
- A measurable **control** $u(t)$ mapping $(0, +\infty)$ into a compact set U
- A stochastic term driven by the coefficient σ

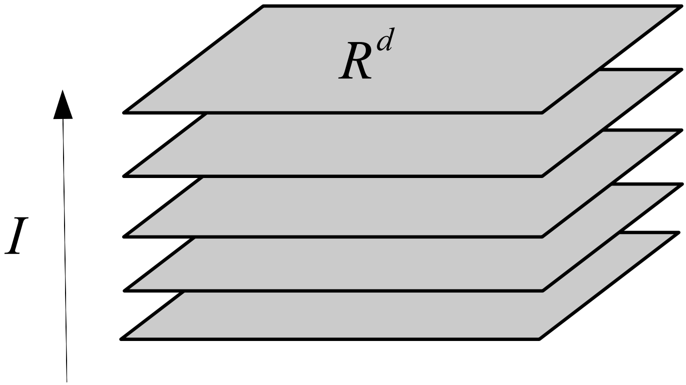
State equation

Evolution for given initial values of X and Q:

$$\begin{cases} dX(t) = f(X(t), Q(t), u(t))dt + \sigma(X(t), Q(t))dW(t), \\ X(0) = x, \\ Q(0) = q. \end{cases}$$

Inside a given set C, **the state may jump** from a state (x, q) to a different state $(x', q') \in D$. The choice of a new state is part of the control strategy

State equations of a hybrid system (2)



The state space is endowed with the **product topology (metric in x, discrete in q)**

Control strategy

A **control** for this hybrid system is a triple:

Control strategy

$$\theta = \left(u, \{\xi_k\}, \left\{(X, Q)\left(\xi_k^+\right)\right\}\right)$$

- u is the controls for the **continuous system dynamics** f
- ξ_k **is a sequence of switching times** for the optional jumps and $(X, Q)(\xi_k^+)$ **are the corresponding states after each jump**

Optimal control problem

Cost functional

In the **discounted infinite horizon case**, the cost functional is defined by

$$
\begin{aligned}
J(x, q, \theta) \quad &= \quad \int_0^{+\infty} \ell(X(t), Q(t), u(t)) e^{-\lambda t} \, dt && (1) \\
&\quad + \sum_{i=0}^{\infty} C(X(\xi_i^-), Q(\xi_i^-), X(\xi_i^+), Q(\xi_i^+)) e^{-\lambda \xi_i} && (2)
\end{aligned}
$$

Optimal control problem

Cost functional

In the **discounted infinite horizon case**, the cost functional is defined by

$$J(x,q,\theta) = \int_0^{+\infty} \ell(X(t),Q(t),u(t))e^{-\lambda t}\,dt \tag{1}$$

$$+\sum_{i=0}^{\infty} C(X(\xi_i^-),Q(\xi_i^-),X(\xi_i^+),Q(\xi_i^+))e^{-\lambda \xi_i} \tag{2}$$

- (1) is the cost related to **continuous control**

Optimal control problem

Cost functional

In the **discounted infinite horizon case**, the cost functional is defined by

$$
\begin{aligned}
J(x,q,\theta) \;&=\; \int_0^{+\infty} \ell(X(t),Q(t),u(t))e^{-\lambda t}\,dt && (1)\\
&\quad + \sum_{i=0}^{\infty} C(X(\xi_i^-),Q(\xi_i^-),X(\xi_i^+),Q(\xi_i^+))e^{-\lambda \xi_i} && (2)
\end{aligned}
$$

- (1) is the cost related to **continuous control**
- (2) is the cost related to **optional (controlled) commutations**

Optimal control problem

Cost functional

In the **discounted infinite horizon case**, the cost functional is defined by

$$J(x,q,\theta) = \int_0^{+\infty} \ell(X(t),Q(t),u(t))e^{-\lambda t}\,dt \quad (1)$$

$$+\sum_{i=0}^{\infty} C(X(\xi_i^-),Q(\xi_i^-),X(\xi_i^+),Q(\xi_i^+))e^{-\lambda \xi_i} \quad (2)$$

- (1) is the cost related to **continuous control**
- (2) is the cost related to **optional (controlled) commutations**

$\lambda > 0$, usual boundedness and Lipschitz continuity assumptions on f, C and ℓ

Bellman Equation (1)

Once defined the value function

$$V(x,q) = \inf_{\theta} \mathbb{E}\big(J(x,q,\theta)\big)$$

it can be proved that (in a suitably adapted viscosity sense) V **satisfies the Quasi-Variational Inequality**

QVI

$$\begin{cases} \max(V(x,q) - \mathcal{N}V(x,q), LV(x,q) + H(x,q,D_x V(x,q)) = 0 & (x,q) \in C, \\ LV(x,q) + H(x, D_x V(x,q)) = 0 & \text{else} \end{cases}$$

Bellman Equation (1)

Once defined the value function

$$V(x,q) = \inf_{\theta} \mathbb{E}\big(J(x,q,\theta)\big)$$

it can be proved that (in a suitably adapted viscosity sense) V **satisfies the Quasi-Variational Inequality**

QVI

$$\begin{cases} \max(V(x,q) - \mathcal{N}V(x,q), LV(x,q) + H(x,q,D_xV(x,q)) = 0 & (x,q) \in C, \\ LV(x,q) + H(x,D_xV(x,q)) = 0 & \text{else} \end{cases}$$

Known results:

- Existence of a viscosity solution
- Strong comparison principle

Value iteration for monotone schemes

"Classical" approach for the approximation: value iteration with **monotone schemes** (e.g., Upwind, Lax–Friedrichs, Semi-Lagrangian + monotone approximation of the switching operators).
Starting from a **time-marching** formulation, the scheme can be put in

Fixed-point form

$$V^h(x,q) = T^h(x,q,V^h) = \begin{cases} \min\left\{N^h V^h(x,q), S^h(x,q,V^h)\right\} & \text{if } x \in C_q \\ S^h(x,q,V^h) & \text{else.} \end{cases}$$

Value iteration for monotone schemes

"Classical" approach for the approximation: value iteration with **monotone schemes** (e.g., Upwind, Lax–Friedrichs, Semi-Lagrangian + monotone approximation of the switching operators).
Starting from a **time-marching** formulation, the scheme can be put in

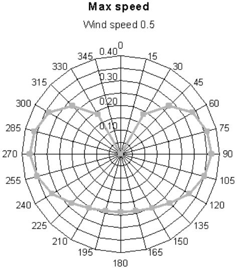
Fixed-point form

$$V^h(x,q) = T^h(x,q,V^h) = \begin{cases} \min\left\{N^h V^h(x,q), S^h(x,q,V^h)\right\} & \text{if } x \in C_q \\ S^h(x,q,V^h) & \text{else.} \end{cases}$$

- The solution can be computed **via the iteration** $V^h_{k+1} = T^h(V^h_k)$
- **Monotone and** L^∞ **stable** under natural assumptions
- From **Barles–Souganidis theorem**, $V^h(x,q) \to V(x,q)$ as $h \to 0$
- Construction of a **quasi-optimal control** from the numerical solution
- Fast solvers via **policy iteration**

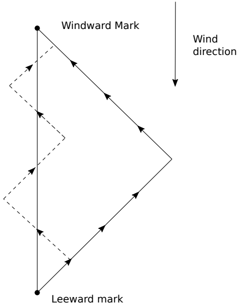
Tacking strategy for a single sailing boat (1)

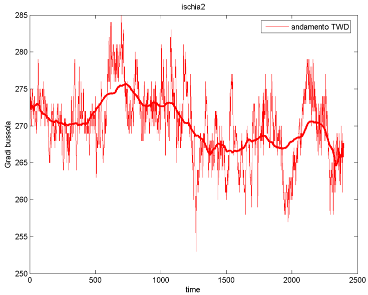
In its most basic form, the *route planning problem* treats the **optimal tacking strategy of a sailing boat** in a windward leg of a regatta.



- The boat sails at about 45^o from the wind direction, which represents the best windward speed obtainable from the **polar plot** of the boat speed w.r.t. the angle with the wind
- Neglecting the **loss of speed in tacking** would result in the **unphysical** possibility of sailing **against the wind**

Tacking strategy for a single sailing boat (2)





The wind direction α has a **partly stochastic evolution**:

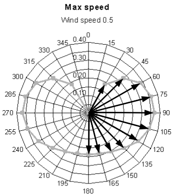
$$d\alpha = c_\alpha dt + \sigma_\alpha dW$$

and its variations should be exploited so as to reach the windward mark in **minimum expected time**

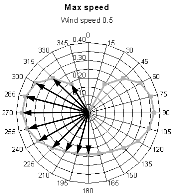
Tacking strategy for a single sailing boat (3)

The **loss of speed** during a change of tack may be modelled as a **switching cost** when jumping between different dynamics

$Q = 1$



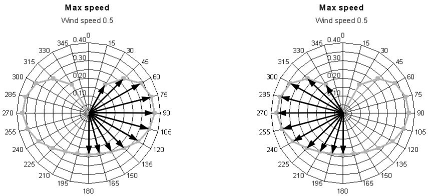
$Q = 2$



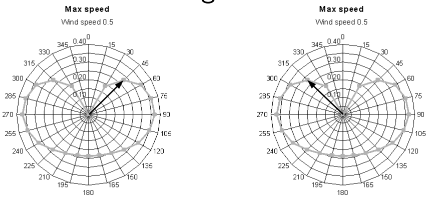
Tacking strategy for a single sailing boat (3)

The **loss of speed** during a change of tack may be modelled as a **switching cost** when jumping between different dynamics

$Q = 1$ $\qquad\qquad$ $Q = 2$



original



simplified

Tacking strategy for a purely windward sailing (1)

Aim: to move in the windward direction as much as possible – in this case, the problem does not depend on the position, but only on the wind direction

- **Cost functional**: discounted **position** + constant **switching cost**

$$J(x, q, \theta) = \int_0^{+\infty} \bar{s} \cos\left(X(t) + \phi_{Q(t)}\right) e^{-\lambda t}\, dt + \sum_{i=0}^{\infty} C e^{-\lambda \xi_i}$$

  with:
  - $X(t) = \alpha(t)$ state variable (wind direction)
  - $\bar{s}$ speed of the boat
  - $\phi_{Q(t)} \approx \pm\pi/4$ angles of the route w.r.t. the wind direction
  - C tacking cost
- **State space**: $\mathbb{R} \times \{1, 2\}$ (wind direction α + boat dynamics (L, R))
- **Heuristics**: *"tacking on a lift"* strategy

Tacking strategy for a purely windward sailing (2)

The resulting **Quasi-Variational Inequality** is in the form

$$\min \left(v(x,q) - v(x,\hat{q}) - C, \lambda v(x,q) - \bar{s}\cos(x+\phi_q) - \frac{\sigma^2}{2}\frac{\partial^2}{\partial x^2} v(x,q) \right) = 0$$

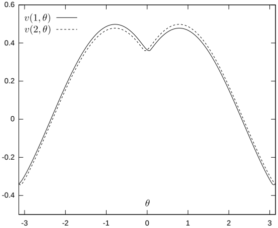
with $\hat{q} \neq q$.

Tacking strategy for a purely windward sailing (2)

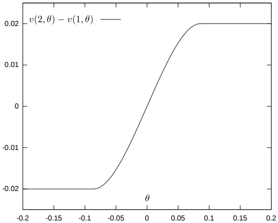
The resulting **Quasi-Variational Inequality** is in the form

$$\min\left(v(x,q)-v(x,\hat{q})-C,\lambda v(x,q)-\bar{s}\cos(x+\phi_q)-\frac{\sigma^2}{2}\frac{\partial^2}{\partial x^2}v(x,q)\right)=0$$

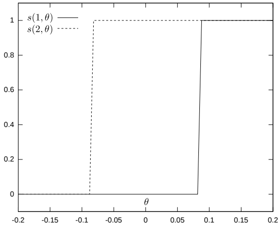
with $\hat{q}\neq q$.
Its solution has the typical behaviour below (semi-explicit solution):



value functions



zoom of the difference



swiching map

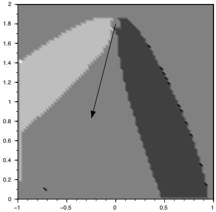
Tacking strategy with a windward target (1)

Cost functional: discounted **minimum time** + **constant cost for controlled switching**

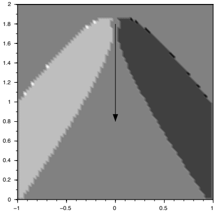
$$J(x, q, \theta) = \int_0^{T_{stop}} e^{-\lambda t}\, dt + \sum_{\xi_i < T_{stop}}^{\infty} C e^{-\lambda \xi_i}$$

- **State space**: $\mathbb{R}^3 \times \{1, 2\}$ (two space dimensions + wind direction + boat direction (L, R))
- **Target problem**: minimum time + penalized distance from the windward mark as a stopping cost
- **Discretization**: SL, $80 \times 80 \times 80$ grid, Modified Policy Iteration
- **Boundary conditions**: state constraints

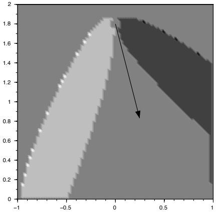
Tacking strategy with a windward target (2)



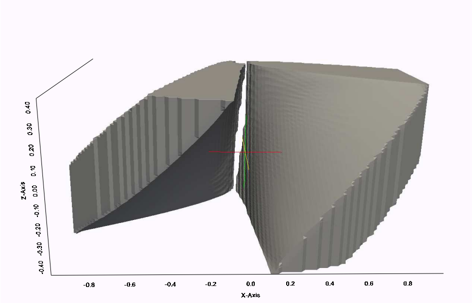
$x_3 = -.25$



$x_3 = 0$



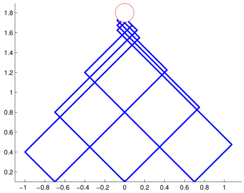
$x_3 = .25$



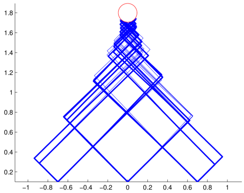
Switching sets, $c_\alpha = 0$

Tacking strategy with a windward target (3)

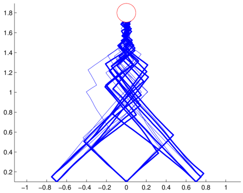
No deterministic drift of the wind ($c_\alpha = 0$), SL discretization as above.
Sample optimal trajectories for increasing variance of the wind direction:



$\sigma_\alpha = 0$



$\sigma_\alpha = 0.01$

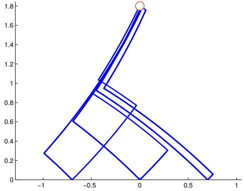


$\sigma_\alpha = 0.1$

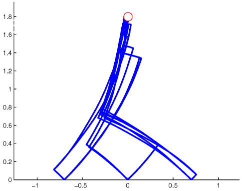
- **Heuristically known**: the tacking region **shrinks** at the increase of wind variance
- At $\sigma_\alpha \approx 0$ the **numerical viscosity** dominates (the effect can be reduced by using the **full dynamics instead of the simplified one**)

Tacking strategy with a windward target (4)

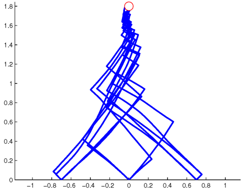
Anti-clockwise drift of the wind ($c_\alpha > 0$), SL discretization as above.
Sample optimal trajectories for increasing variance of the wind direction:



$\sigma_\alpha = 0$



$\sigma_\alpha = 0.05$



$\sigma_\alpha = 0.1$

- **Heuristically known**: the optimal strategy tends to keep the trajectory **on the left side of the state space**
- For increasing σ_α this strategy is **blended with the previous one**

Tacking strategy in a match race (1)

Tacking strategy in a match race (2)

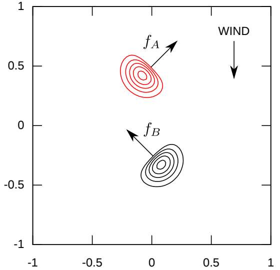
Aim: be **ahead** of the other player – as in a **pursuit–evasion** game

Each of the players wants to **avoid the turbulent region below the other player**, and vice versa each of the two wants to exploit this region to **slow down the other one** (video)

Tacking strategy in a match race (2)

Aim: be **ahead** of the other player – as in a **pursuit–evasion** game
Each of the players wants to **avoid the turbulent region below the other player**, and vice versa each of the two wants to exploit this region to **slow down the other one** (video)

Dynamics: both players follow the dynamics of a single boat, but there exists an **influence** between the two:



The turbulence generated by a player is modelled as a region of **reduced speed for the other**

Tacking strategy in a match race (3)

- **Wind dynamics**: purely Brownian
- **Cost functional**: discounted **difference for the component** X_2 + no autonomous switching + **constant cost for controlled switching**

$$J(x, q, \theta^A, \theta^B) = \int_0^{+\infty} \left(X_2^A(t) - X_2^B(t) \right) e^{-\lambda t} \, dt + \sum_{i=0}^{\infty} C^B e^{-\lambda \xi_i^B} - \sum_{i=0}^{\infty} C^A e^{-\lambda \xi_i^A}$$

- **State space**: $\mathbb{R}^3 \times \{1, 2, 3, 4\}$ (two space dimensions + wind direction + both boat directions (LL, LR, RL, RR)). Use of **reduced coordinates** as in a pursuit–evasion game
- **Aim**: being **as windward as possible** w.r.t. the other player:
  $A \to \max J$,
  $B \to \min J$
- Use of the one-dimensional problem to provide **boundary conditions** for the value function

Tacking strategy in a match race (4)

Value functions: in principle,

$$\sup_{\theta^A} \inf_{\theta^B} \neq \inf_{\theta^B} \sup_{\theta^A}$$

so we should consider Upper and Lower Value Functions (in the sense of non-anticipative strategies by Elliot-Kalton):

$$V^-(x,q) = \inf_{\theta^B} \sup_{\theta^A} \mathbb{E}\left(J(x,q,\theta^A,\theta^B) \right)$$

$$V^+(x,q) = \sup_{\theta^A} \inf_{\theta^B} \mathbb{E}\left(J(x,q,\theta^A,\theta^B) \right)$$

- Each of the two value functions may be characterized via a suitable **quasi-variational inequality**
- Technical conditions (**"no free loop condition"**) for obtaining a **comparison lemma**, and hence uniqueness. If a suitable extended Isaacs' condition is satisfied, then **the game has a value** (this seems to be the case from numerical simulations)

A test in asymmetric conditions

Both players have the **same speed**. The red player **leads at the start, but has a higher switching cost**. The black player **exploits better the wind variations** and eventually passes the other one.

Computational issues (1)

- **Numerical examples** carried out on a Lenovo Ultrabook X1 Carbon (4 cores, i5, 1.9 GHz), C++/OpenMP code

Computational issues (1)

- **Numerical examples** carried out on a Lenovo Ultrabook X1 Carbon (4 cores, i5, 1.9 GHz), C++/OpenMP code
- **First-Order upwind** scheme for the QVI, first attempts with value iteration (or modified policy iteration for the one-player case), warm start for the game
- **Boundary conditions:** penalization (state constraints) for the one-player case, **decoupled game** for the Isaacs' case

Computational issues (1)

- **Numerical examples** carried out on a Lenovo Ultrabook X1 Carbon (4 cores, i5, 1.9 GHz), C++/OpenMP code
- **First-Order upwind** scheme for the QVI, first attempts with value iteration (or modified policy iteration for the one-player case), warm start for the game
- **Boundary conditions:** penalization (state constraints) for the one-player case, **decoupled game** for the Isaacs' case
- Up to $\mathbf{3.2 \cdot 10^7}$ DOF handled
- **OpenMP** parallelization suffers from heavy data exchange. With a $100 \times 100 \times 100$ grid:

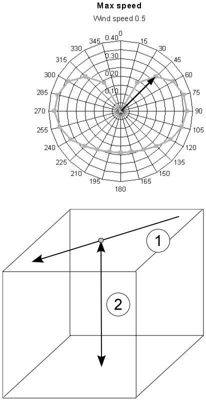
Threads	CPU time
1	618.2
2	351.7
4	279.3

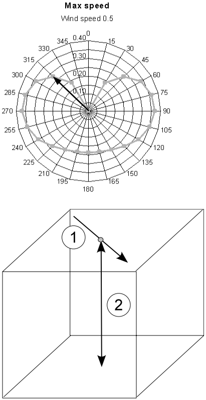
Computational issues (2)

Further attempt: Fast sweeping, but with a decoupling of the diffusive part

Computational issues (2)

Further attempt: Fast sweeping, but with a decoupling of the diffusive part





1. Sweep against the dynamics
2. Exact solver for the diffusion in the vertical direction

Computational issues (3)

- IT: pure value iteration (no adapted order for the variables)
- FS-IT: Fast Sweeping + iterative solution of the diffusion term
- FS-LU: Fast Sweeping + LU solution of the diffusion term (LAPACK routines DGTTRF for tridiagonal LU factorization + DGTTRS tridiagonal solver)

Method	$\sigma = 0$	$\sigma = 0.01$	$\sigma = 0.025$	$\sigma = 0.05$
IT	185s (286)	243s (374)	558s (852)	1577s (2412)

Computational issues (3)

- IT: pure value iteration (no adapted order for the variables)
- FS-IT: Fast Sweeping + iterative solution of the diffusion term
- FS-LU: Fast Sweeping + LU solution of the diffusion term (LAPACK routines DGTTRF for tridiagonal LU factorization + DGTTRS tridiagonal solver)

Method	$\sigma = 0$	$\sigma = 0.01$	$\sigma = 0.025$	$\sigma = 0.05$
IT	185s (286)	243s (374)	558s (852)	1577s (2412)
FS-IT	5.9s (12)	39s (79)	160s (326)	550s (1119)

Computational issues (3)

- IT: pure value iteration (no adapted order for the variables)
- FS-IT: Fast Sweeping + iterative solution of the diffusion term
- FS-LU: Fast Sweeping + LU solution of the diffusion term (LAPACK routines DGTTRF for tridiagonal LU factorization + DGTTRS tridiagonal solver)

Method	$\sigma = 0$	$\sigma = 0.01$	$\sigma = 0.025$	$\sigma = 0.05$
IT	185s (286)	243s (374)	558s (852)	1577s (2412)
FS-IT	5.9s (12)	39s (79)	160s (326)	550s (1119)
FS-LU	6.3s (9)	7.4s (14)	6.9s (13)	6.8s (13)

CPU time (iteration number) for the various solvers
$100 \times 100 \times 100$ nodes, stopping tolerance $\varepsilon = 10^{-8}$

Final remarks

- **Sound theoretical framework**, for both the theoretical and the computational aspects
- **Viable and robust design** of a feedback controller in a feasible dimension of the state space
- Possibility of using **acceleration techniques** of Policy Iteration or Fast Sweeping type in the one-player setting
- Heuristically known qualitative features of optimal solutions are **well reproduced**
- Open problems:
  - **Comparison principle** for the Isaacs' system in the symmetric case (i.e., in lack of the **"no free loop condition"**)
  - Suitable definition and convergence of **(modified) policy iteration** in the two-player setting
- **Planned improvement:** target problem for the game (5-d, no use of reduced coordinates)

References

- A. Bensoussan and J.L. Menaldi, *Hybrid control and dynamic programming*, Dynam. Contin. Discrete and Impuls. System, **3** (1997), 395–442
- M.S. Branicky, V. Borkar and S. Mitter, *A unified framework for hybrid control problems*, IEEE Trans. Autom. Contr., **43** (1998), 31–45
- S. Cacace, R. Ferretti and A. Festa, *Hybrid differential games and their application to a match race problem*, Appl. Math. Comp. (to appear)
- S. Dharmatti and M. Ramaswamy, *Hybrid control system and viscosity solutions*, SIAM J. Contr. Optim., **34** (2005), 1259–1288
- B. El Asri and S. Mazid, *Stochastic differential switching game in infinite horizon*, ArXiv preprint
- R. Ferretti and A. Festa, *Optimal route planning for sailing boats: a hybrid formulation*, J. of Optim. Theory and Applications, **181** (2019), 1015–1032
- R. Ferretti and A. Sassi, *A semi-Lagrangian algorithm in policy space for hybrid optimal control problems*, ESAIM: COCV **24** (2018), 965–983
- R. Ferretti and H. Zidani, *Monotone numerical schemes and feedback construction for hybrid control systems*, J. of Optim. Theory and Applications **165** (2014), 507–531