Stochastic Control: from Gradient Methods and Dynamic Programming to Statistical Learning

Gilles Pagès, Olivier Pironneau

Stochastic Control: from Gradient Methods and Dynamic Programming to Statistical Learning

Gilles Pagès[1], Olivier Pironneau[2]

Abstract

In this article the authors wish to contribute to the evaluation of statistical learning for stochastic control. We will review the well known methods for stochastic control and compare their numerical performance to those of a neural network. This will be done on a simple but practical example arising for fishing quotas to preserve the biomass of fish.

Keywords: Stochastic control, Dynamic Programming, Partial Differential Equations, Statistical Learning.

1. Introduction

Stochastic control of differential systems with random data or modelling errors, is usually solved by dynamic programming, a technique introduced in the 1950 by Richard Bellman[1]. Numerically one has to solve a backward in time differential equation possibly coupled with the forward stochastic differential equation (SDE) of the problem. The method has various discrete implementations (see [3],[14], etc). Calculus of variations and optimal control is another approach, which, in principle is more expensive because the backward adjoint Partial Differential Equation (PDE) and the forward Kolmogorov PDE for the PDF of the process are coupled and must be solved iteratively by a gradient method.

A third method has been proposed in [2], [5], [4], [6] and others : statistical learning with neural networks: like off-line/on-line reduction methods, one is ready to pay a high CPU price in a "learning" phase so as to solve quickly the problem with new data with the "trained network". The novelty is that a trained neural network can be ported on a small unit like a smart phone and yet be fast and accurate.

In this article the three classes of methods will be tested on an easy yet practical problem studied in [8]: the control of a fishing site in Senegal where the fish biomass is threatened by intensive fishing. In [6] it was shown that parameter identification from a few measures near time zero could be obtained accurately and fast by statistical learning. Here the parameters are known but with a level of uncertainty and the objective is to apply optimal quotas to stabilize the fish biomass at a desired level in this stochastic setting.

The model has two ordinary differential equations (ODE), one for the fish biomass X_t at time t, and one for the fishing effort which, for us, will be the number of boats at sea E_t.

[1] *gilles.pages@sorbonne-universite.fr*, LPMA, Sorbonne Université, Paris, France

[2] *olivier.pironneau@sorbonne-universite.fr* , LJLL, Sorbonne Université, Paris, France.

Quotas are meant to keep the biomass at a constant optimal level. Hence the total number of fish caught must be lower than the quota Q_t. Then each boat is told to keep their catch below Q_t/E_t. By construction each boat can't catch more than qX_t, where q is a constant reflecting the structural limit of the fisherman's catch yet proportional to the biomass.
The control problem is solved by minimizing a cost function, the integral over time of $\|X_t - X_d\|^2$ where X_d is the desired biomass. But quotas, imposed by the regulator, are unpopular. To avoid too strict quotas, we add to the cost function the integral over time of $-\alpha Q_t$ where α is constant; consequently, a small quota will not be optimal because it gives a larger value to the cost function.
Along the way we found that the solution of the problem is very often "bang-bang", i.e; switching from the structural bound to the quota often. For the regulator, it is also unpopular; hence in an attempt to avoid such frequent drastic changes we add to the cost function the integral over time of $\|\dot{Q}_t\|^2$.
As the core of the article is to compare various numerical solution method, the detail of the modelling has been postponed to Appendix A.
The plan is as follows:
In section one the fishing quota control problem is stated. It is a stochastic control problem because the data are not known accurately. Existence of solutions and discretization methods are discussed. We identify 3 classes: deterministic control (no noise), stochastic state equation but deterministic control (which we call non-dynamic) and fully stochastic state and control, known as dynamic programming.
In section 2, a Monte-Carlo method is proposed for the non-dynamic case. It is a brute force Monte-Carlo approach, much more expensive than the Dynamic Programming approach of section 4, but it has an advantage: it can be ported to a Neural Network for training and testing avoiding also recurrent networks, inevitably less accurate.
Finally, in the last section PDE methods are investigated, based either on Ito Calculus or Kolmogorov equations for the probability density function (PDF) of the process.

2. A model problem

To study the evolution of the biomass of a fishing site we consider the following stochastic optimal control problem

$$\min_{u\in\mathcal{U}} J(u) \;:= \int_0^T \mathbb{E}\left[(X_t - X_d)^2 - \alpha u(X_t,t) + \beta|\partial_t u(X_t,t)|^2\right] \mathrm{d}t, \tag{1}$$

$$\text{with} \quad \mathrm{d}X_t \;= X_t(r - \kappa X_t - u_t)\mathrm{d}t + X_t\sigma \mathrm{d}W_t,\; X_0 = X^0 \geq 0, \tag{2}$$

where W_t is a $\mathcal{N}_{0,1}$ Weiner process; $X_d, r, \kappa, \sigma, u_m, u_M$ are real constants which depend on the fishing site considered, and

$$\mathcal{U} = \{u \in H^1(]0,T[) \;:\; u_m \leq u(X_t,t) \leq u_M, a.e.\} \tag{3}$$

This problem arises for the determination of fishing quotas (see Appendix A for details): X_t is the fish biomass at time t and $u_t X_t$ is the fishing quota at time t. Occasionally we will also display a variable proportional to the number of boats at sea, the "fishing effort" $E(t)$, modelled by

$$\mathrm{d}E_t = (a - u_t X_t - cE_t)\mathrm{d}t, \quad E_0 = E^0 \text{ given.}$$

where a is related to the price of fish and c is the operating cost of the fisherman. As explained in Appendix A, $a - u_t X_t$ is the profit of the fisherman and it decreases when too many fish are caught because the price of fish collapse.
The control problem aims at keeping a biomass at an desired level X_d at all times, but at the same time, quotas being unpopular, there is a term to prevent small quotas, $-\alpha u_t$ and another to prevent quotas from changing too quickly: $\beta|\partial_t u|^2$. Note that $(X,t) \mapsto u(X,t)$ is a deterministic function and $\{u(X_t,t)\}_t$ is a Markov process. Finally, $\partial_t u(X_t,t)$ means $\partial_t u(X,t)|_{X=X_t}$.
It is important to notice that one may also consider the case where u is a deterministic function of t only:

$$\min_{u\in\mathcal{U}} J(u) := \int_0^T \mathbb{E}\left[(X_t - X_d)^2\right] - \alpha u(t) + \beta|\partial_t u(t)|^2\right] \mathrm{d}t, \text{ with (2).} \tag{4}$$

We shall refer to it as the non-dynamic problem by opposition to dynamic programming where $u(X_t,t)$ is stochastic because of X_t but Markovian. In the non-dynamic case, the optimal control is the same for all trajectories X_t and in the dynamic case u adapts to the trajectories. To our surprise the non-dynamic case turns out to be more precise and more suitable because the day-to-day noise may not be very meaningful.
Occasionally we shall also refer to the deterministic case, i.e. when $\sigma = 0$.

2.1. Existence of solution: the deterministic case
In the deterministic case the problem is

$$\min_{u\in\mathcal{U}}\{J(u) = \int_0^T |X - X_d]^2 - \alpha u + \beta|\dot{u}|^2 \ : \ \dot{X} = X[(r - \kappa X - u),\ X(0) = X^0\} \tag{5}$$

As u is positive $0 \le X(t) \le X_M := X^0 e^{rT}$. Then obviously $F(X) := rX - \kappa X^2 - Xu$ is uniformly Lipschitz in X:

$$|F(X_1(t)) - F(X_2(t))| \le |X_1(t) - X_2(t)|(r - u_m + 2\kappa X_M|,\quad \forall t \in [0,T];$$

therefore the solution of the ODE exists and is unique. It is also uniformly Lipschitz continuous in u:

$$\begin{aligned}
&\dot{X}_i = rX_i - \kappa X_i^2 - u_i X_i,\ i = 1,2,\ \Rightarrow\\
&\dot{X}_1 - \dot{X}_2 = (X_1 - X_2)\,(r - \kappa(X_1 + X_2) - (u_1 + u_2)) - (u_1 - u_2)(X_1 + X_2),\ \Rightarrow\\
&|\dot{X}_1 - \dot{X}_2| \le |X_1 - X_2|(r + 2X_M + 2u_M) + 2X_M|u_1 - u_2|,\ \Rightarrow\\
&|X_1 - X_2| \le 2|u_1 - u_2|X_M(\exp((r + 2X_M + 2u_M)t) - 1)/(r + 2X_M + 2u_M)
\end{aligned} \tag{6}$$

where the last line is due to the Grönwall lemma. This shows also that $u \mapsto \int_0^T |X(t) - X_d|^2 \mathrm{d}t$ is continuous in $L^2(]0,T[)$.

Proposition 1. *Problem (5) has a solution.*

Proof: Consider a minimizing sequence: $u^n \in \mathcal{U}$, $J(u^n) \to \inf_{u\in\mathcal{U}} J(u)$. $\mathcal{U}$ being non-empty, bounded and closed and $\|\dot{u}\|_0^2$ being part of the cost function, there is a subsequence with $u^n \to u^* \in \mathcal{U}$ weakly in $H^1(]0,T[)$ and strongly in $L^2(]0,T[)$. By the above established continuity, $X^n \to X^*$ in $L^2(]0,T[)$. Obviously $\int_0^T \alpha u^n \to \int_0^T \alpha u^*$ and by the lower semicontinuity of $u \to \|\dot{u}\|_0^2$,

$$J(u^*) \le \liminf J(u^n) = \inf_{u\in\mathcal{U}} J(u).$$

Therefore u^* is a minimum and problem (5) has a solution.

2.2. Existence of solution: the stochastic non-dynamic case

Proposition 2. *The solution of of the SDE (2) exists and is positive.*

Proof: $X \to (r - \kappa X - u_t)X$ is locally Lipschitz, uniformly in t since $u_t \in [u_m, u_M]$; hence for every realization $W_t(\omega)$ there is a unique strong solution until a blow-up time τ which is a stopping time for the filtration $\mathcal{F}_t^w = \underline{\sigma}(W_s, s \geq t, \mathcal{N}_s)$. On $[0, \tau[$ we may work with $Y_t = \log X_t$, solution (by Itô calculus) of

$$\begin{aligned} \mathrm{d}Y_t &= (r - \frac{\sigma^2}{2} - \kappa X_t - u_t)\mathrm{d}t + \sigma \mathrm{d}W_t, \\ &\Rightarrow \; X_t = X^0 \exp\left((r - \frac{\sigma^2}{2})t - \int_0^t (\kappa X_s + u_s)\mathrm{d}s + \sigma W_t\right). \\ &\Rightarrow \; X_t \leq X^0 e^{S_t(u_m)}, \forall t \in (0, \tau], \quad \text{where } S_t(v) := (r - \frac{\sigma^2}{2})t - vt + \sigma W_t. \end{aligned} \tag{7}$$

Hence $\int_0^\tau X_s \mathrm{d}s = +\infty$ is impossible unless $\tau = +\infty$, $\mathcal{P}$ a.s. Therefore $X_t \in \mathbb{R}^+, \forall t$ and

$$X^0 e^{S_t(u_M)} \exp\left(-\kappa \int_0^t e^{S_s(u_M)} \mathrm{d}s\right) \leq X_t \leq X^0 e^{S_t(u_m)}. \tag{8}$$

Theorem 1. *Problem (1)(2) has a solution.*

Proof: Consider two trajectories corresponding to the same event but different u, denote δ the difference and $\bar{}$ the average; we have

$$\mathrm{d}\delta X_t = [\delta X_t (r - 2\kappa \bar{X}_t - \bar{u}) - \bar{X}_t \delta u]\mathrm{d}t + \sigma \delta X_t dW_t.$$

By Itô calculus:

$$(\delta X_t)^2 = 2\int_0^t [\delta X_s)^2 (r + \frac{\sigma^2}{2} - 2\kappa \bar{X}_t - \bar{u}) - \bar{X}_s \delta u(s) \delta X_s] \mathrm{d}s + 2\int_0^t \sigma (\delta X_s)^2 \mathrm{d}W_s.$$

Hence

$$\begin{aligned} \mathbb{E}[(\delta X_t)^2] &\leq 2(r + \frac{\sigma^2}{2} - u_m) \int_0^t \mathbb{E}[(\delta X_s)^2] - \int_0^t \delta u(s) \mathbb{E}[\bar{X}_s \delta X_s] \mathrm{d}s \\ &\leq 2(r + \frac{\sigma^2}{2} - u_m) \int_0^t \mathbb{E}[(\delta X_s)^2] + \int_0^t \delta u(s) \mathbb{E}[(\bar{X}_s)^2] \mathrm{d}s. \end{aligned} \tag{9}$$

We have seen earlier that $0 \leq X_t \leq X_0 e^{(r - \frac{1}{2}\sigma^2)t + \sigma W_t}$, therefore $E[(\bar{X}_s)^2] \leq 4e^{(2r + \sigma^2)s} \mathbb{E}[X_0^2]$, so much so that by Grönwall's lemma,

$$\mathbb{E}[(\delta X_t)^2] \leq 2(r + \frac{\sigma^2}{2} - u_m) \int_0^t \mathbb{E}[(\delta X_s)^2] \mathrm{d}s + 4e^{(2r + \sigma^2)T} \mathbb{E}[X_0^2] \int_0^t (\delta u(s))^2 ds;$$

and by Grönwall's lemma again,

$$\mathbb{E}[(\delta X_t)^2] \leq 4e^{2r + \sigma^2 - 2u_m)t} e^{(2r + \sigma^2)T} \mathbb{E}[X_0^2] \int_0^t |\delta u(s)| ds.$$

Consequently, for some $C \in \mathbb{R}^+$ function of r, σ, u_m, T only,

$$\sup_{t \in [0,t]} \mathbb{E}[(\delta X_t)^2] \leq C \mathbb{E}[X_0^2] \|\delta u\|_{L^1(]0,T[)}$$

The continuity of $u \to \int_0^T \mathbb{E}[(X_t - X_d)^2] \mathrm{d}t$ follows and existence is then shown as for the deterministic case.

2.3. Existence of solution: the dynamic case

To establish existence by probabilistic methods proved to be rather hard. On the other hand, the problem can be converted into an equivalent distributed control problem by using the Kolmogorov equation of the PDF of X_t.

We will show in section 7 that the problem has a solution when the expected value in the cost function and κ are multiplied by a cutoff function for large X and

$$\mathcal{U} = \{u \in L^2\left(W^{1,1}(\mathbb{R}^+)\right) : \ u_m \le u \le u_M, \ \partial_t u \in L^2(\mathcal{Q}^+), \ \lim_{Y\to\infty} |\log X| u(X,t) \le \infty.\}$$

2.4. Discretization

Let $h = \frac{T}{M}$ and $t^m = mh$. For any f, f^m denotes an approximation of $f(mh)$.

Define (Euler scheme)

$$X^{m+1} = X^m\left(1 + h(r - \kappa X^m - u^m) + \sigma\delta W^m\right) \tag{10}$$

where $\delta W^m = W_{(m+1)h} - W_{mh}$. Note that positivity may not be preserved by this scheme, but we may use (7):

$$X^{m+1} = X^m \exp\left((r - \frac{\sigma^2}{2})h - (\kappa X^m + u^m)h + \sigma\delta W^m\right).$$

Alternatively, to avoid computing exponentials, we may consider the Milstein scheme:

$$X^{m+1} = X^m\left(1 + h(r - \kappa X^m - u^m - \frac{\sigma^2}{2}) + \sigma\delta W^m + \frac{\sigma^2}{2}(\delta W^m)^2\right). \tag{11}$$

Proposition 3. *Let $\bar{X}$ denote the interpolated Milstein scheme:*

$$\bar{X}_t = X^m\left(1 + (t - mh)(r - \kappa X^m - u^m - \frac{\sigma^2}{2}) + \sigma(W_t - W^m) + \frac{\sigma^2}{2}((W_t - W^m)^2\right).$$

If $\sigma^2 \le 1 + h(r - \kappa\bar{X}_t - u_M)$, then $\mathbb{E}[\bar{X}_t] \ge 0$.

Proof: The above inequality implies that the discriminant of the second degree polynomial in δW is negative: $\sigma^4 - \sigma^2(1 - 2h(r - \kappa X^m - u^m)) \le 0$.

3. Brute force solution of the non-dynamic control problem by Monte-Carlo

Recall that numerically $\delta W^m = \sqrt{h}\mathcal{N}_{0,1}^m$ where $\mathcal{N}_{0,1}^m$ denotes the numerical realization at time step m of a normal Gaussian random variable. For each of the 3 previous schemes there is an ζ_h such that

$$X^{m+1} = \zeta_h(X^m, u^m, \mathcal{N}_{0,1}), \ X^0 \text{ given}, \tag{12}$$

Let $\{X_k^m\}_1^K$ be K realizations of (12); by the law of large numbers :

$$\mathbb{E}[(X_t - X_d)^2]_{|mh} \approx E_K[(X^m - X_d)^2] := \frac{1}{K}\sum_1^K (X_k^m - X_d)^2.$$

Problem (1) is approximated by

$$\min_{u_m \le u^m \le u_M} J_K := h\sum_0^{M-2}\left[E_K[(X^{m+1} - X_d)^2] - \alpha u^m + \frac{\beta}{h^2}|u^{m+1} - u^m|^2\right], \text{ with (12)}. \tag{13}$$

Note that $\{u^m\}$ need not be Markovian; it may depend also on X^p, $p > m$.

3.1. Solution with a gradient method

Gradient methods update $\{u^m\}_1^{M-1}$ by $u^m \leftarrow u^m - \mu G^m$, where μ, the step size, is an appropriate scalar and $G^m = \partial_{u^m} J_K$. Let X_k^m be computed by (10); let us differentiate this equation and multiply it by an adjoint variable P_k^m and sum:

$$\sum_{m=0}^{M-2} \left[-P_k^m \delta X_k^{m+1} + P_k^m \left(\delta X_k^m \left(1 + h(r - \kappa X_k^m - u^m) + \sigma \delta W_k^m\right) - X_k^m h(\kappa \delta X_k^m + \delta u^m)\right)\right] = 0.$$

As $\delta X^0 = 0$, if we let $P^{M-1} = P^{M-2} = 0$, and define

$$v_k^m = 1 + h(r - 2\kappa X_k^m - u^m) + \sigma \mathrm{d}W_k^m, \text{ then}$$
$$-\sum_0^{M-2} h P_k^m X_k^m \delta u^m = \sum_1^{M-2} (P_k^{m-1} \delta X_k^m - P_k^m \delta X^m v_k^m) = \sum_1^{M-2} \delta X_k^m (P_k^{m-1} - P_k^m v_k^m). \quad (14)$$

Let us define P_k^m by

$$P_k^{m-1} = P_k^m v_k^m - 2(X_k^m - X_d^m), \quad m = M-2, M-3, .., 1. \text{ Then}$$
$$\sum_{m=1}^{M-1} \delta E_K[(X^m - X_d)^2] = \frac{2}{K} \sum_{k=1}^{K} \sum_{m=1}^{M-1} (X_k^m - X_d) \delta X_k^m$$
$$= -\frac{1}{K} \sum_{k=1}^{K} \sum_{m=1}^{M-1} (P_k^{m-1} - P_k^m v_k^m) \delta X_k^m = \frac{1}{K} \sum_{k=1}^{K} \sum_{m=0}^{M-2} P_k^m X_k^m \delta u_k^m. \quad (15)$$

So the discrete gradient is

$$G^m = \frac{1}{K} \sum_1^K P_k^m X_k^m - \alpha - 2\frac{\beta}{h^2}(u^{m+1} - 2u^m + u^{m-1}).$$

3.2. Implementation and examples

We use the `L-BFGS-B` function of the Python library `minimize`. It is an accelerated projected gradient method for problems with box constraints. We took 50 time steps and chose to solve a deterministic case with

$$\sigma = 0, K = 1, X_d = 1, \kappa = 1.2, X_0 = 0.9, c = 1, a = 1, u_m = 0.5, \ u_M = 1. \quad (16)$$

The computer program is in file `fishingQuotaBruteforce.py`. The results with $\alpha = \beta = 0$ are shown on the left in figure 1 with a saturation of the constraints $u = u_m$ when $m < 7$; but the solution is not bang-bang everywhere and $u = 0.8$ gives $X^{m+1} = X^m$. On the right in figure 1 the results are shown for $\alpha = 0.01$, $\beta = 0.001$. On figure 2 a stochastic case of (13) is solved with the same parameters (16) except $\sigma = 0.3$. The results are shown for $K = 1, 10, 100, 10000$ Monte-Carlo samples.
Note that the brute force solution for $K >> 1$ is quite close to the noiseless solution of figure 1-b.

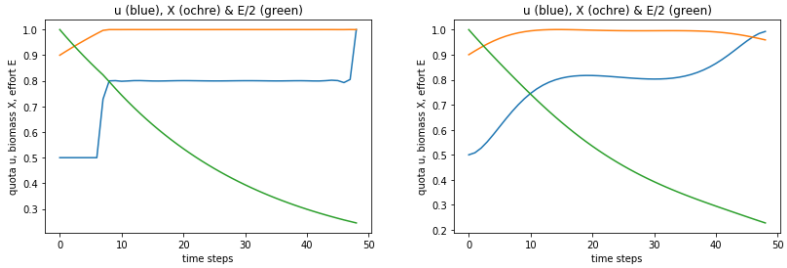

Figure 1: Solution of (1) discretized by (13), versus m, in the deterministic case ($\sigma = 0, K = 1$) with $\beta = 0$ (left), and $\beta = 0.001$ (right).The blue curve is the control u; the biomass X_t is in ochre and the fishing effort in green.

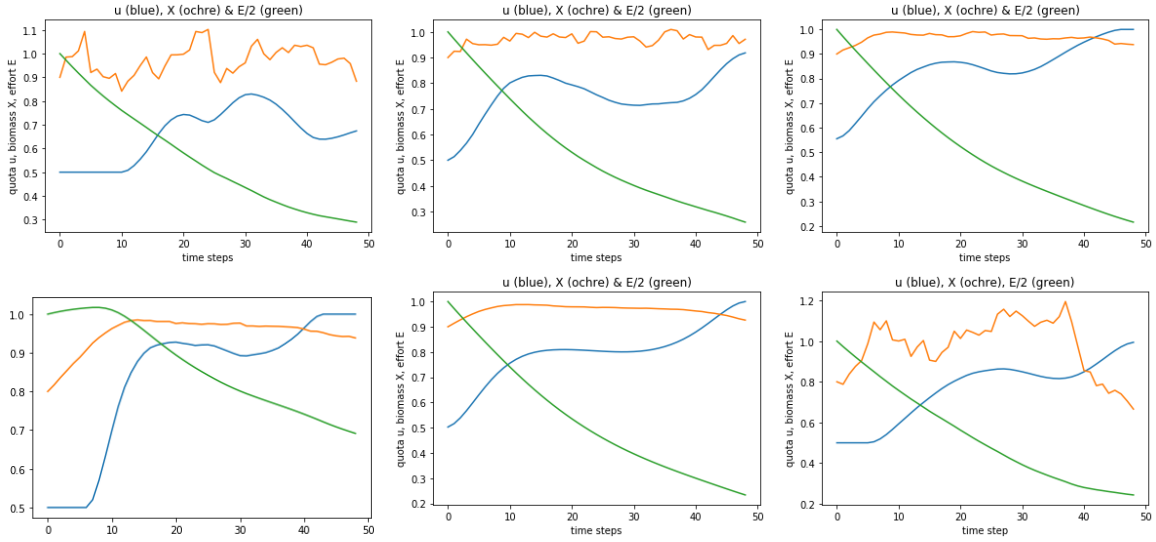

Figure 2: Solution of (13) in the stochastic case with $\sigma = 0.3$: convergence of the Monte-Carlo approximation . Left to right, up first: K=1, K=10, K=100,K=1000, K=10000. Already the results are fairly converged at K=100. In the last figure the optimal control is computed with K=100 and used to solve (2) with a new random noise, thereby showing that the solution is still acceptable.

4. Solution of the non-dynamic control problem by statistical learning

To reduce the CPU cost of solving

$$\min_{u\in\mathcal{U}} J(u) \;:= \int_0^T [\frac{1}{K}\sum_0^K (X_k(t)-X_d)^2 - \alpha u(t) + \beta(\partial_t u(t))^2]\mathrm{d}t \tag{17}$$

$$\text{with} \quad \mathrm{d}X_k \;= X_k(t)((r-\kappa X_k(t)-u(t))\mathrm{d}t + \sigma \mathrm{d}W_k(t), \quad X(0)=X^0, \tag{18}$$

we shall train a neural network with input parameters $[X^0, r, \kappa]$ and output an optimal $\{u^m\}_1^{M-1}$ for these parameters, computed by the brute force method above. There is no need to vary the parameter X_d because we can always return to the case $X_d = 1$ by scaling.
To build a training set we use a large number of synthetic solutions obtained with the brute force method of the previous section. However as noise is known not to affect neural network we use a low value $K = 100$ to generate 10000 solutions (it takes about 90 minutes on an intel core i7 machine). It works as follows:

- Choose $[X^0, r, \kappa]$ uniformly random in $[X^0_m, X^0_M] \times [r_m, r_M] \times [\kappa_m, \kappa_M]$.
- Generate $K \cdot M$ normal gaussian random values for $\mathrm{d}W_t$.
- Compute a discrete optimal solution $\{u^m\}_1^{M-1}$ of (17) with K=100 by the L-BFGS-B Python module.
- Then train a neural network $[X^0, r, \kappa] \to \{u^m\}_1^{M-1}$ with these 10000 synthetic solutions.
- Finally check the precision by using the network's answer $\{u^m\}_1^{M-1}$ to a new set of values for $[X^0, r, \kappa]$ and compare with the brute force solution corresponding to these parameters .

For an introduction to Statistical Learning with Neural Networks see [12]. We have used `Keras` (see [10]) to generate a 4-layer Neural Network (see figure 3). The input layer has 3 inputs; the two hidden layers have 50 neurons each and the output layer has M nodes. All layers use the ReLU activation. The two hidden layers are separated by a batch-normalisation.

4.1. A numerical test

We have taken 50 time steps and K=100 for the brute force part. The following parameters are used:

$$T = 2, \alpha = 0.01, \beta = 0.001, \sigma = 0.3, X_d = 1, u_m = 0.5, u_M = 1$$

For the Neural Network training we have used 10000 samples in batches of 32 and a maximum of 2000 epochs. The input parameters are

```
r = np.random.uniform(1.9,2.2)
kappa = np.random.uniform(1,1.3)
X0=np.random.uniform(0.7,1.3)
```

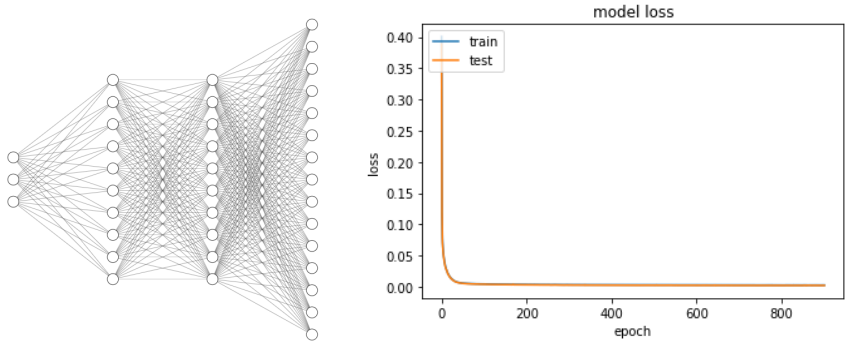

Figure 3: The Neural Network has 3 inputs, two hidden layers each with 50 neurons (10 shown here) and separated by batch-normalisation and an output layer with M output (M=15 shown here) . Right: typical convergence curve of the loss function during the training phase.

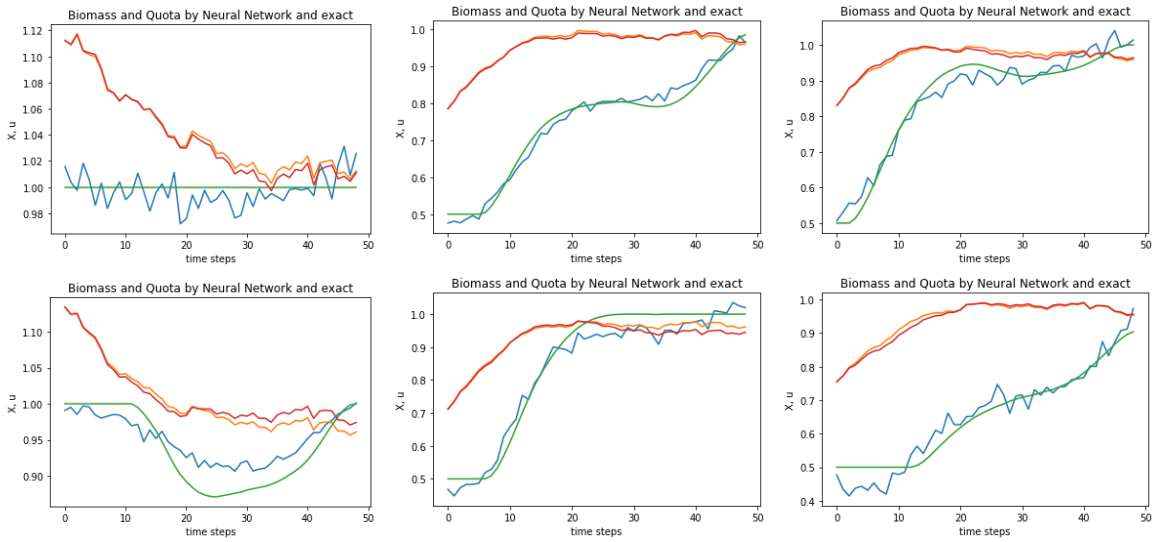

Figure 4: On each of the 6 plots X and u are shown: True discrete solution in green for X and red for u. NN solution in blue for X_{NN} and ochre for u_{NN}. The plots correspond to the following values of the parameters: $r = 2.0228569911704177, \kappa = 1.0203759108766386, X_0 = 1.1122268356898797$, $r = 2.0688190264082973, \kappa = 1.2641632371604865, X_0 = 0.7852264857858362$, $r = 1.9151933649036637, \kappa = 1.0141074063562392, X_0 = 0.8299957532445648$, $r = 2.0738603520220655, \kappa = 1.187620910330912, X_0 = 1.1344043926097065$, $r = 2.0463252714459936, \kappa = 1.122409869546898, X_0 = 0.7116598707629698$, $r = 1.9002636936181148, \kappa = 1.189933237216782, X_0 = 0.7550552945829774$.

Finally for the fishing effort equation, $a = c = 1$.
The computer program is written in Python and is in the file `fishingQuotaIA4.py`. The learning phase produced the following precision on each of the time steps :

```
Test loss: 0.03751661651
Average relative error |1-u_NN/u| for the 49 time steps =
[0.03185646 0.03954296 0.04612997 0.03440035 0.03665108 0.04323719 0.03549499 0.04364510
 0.04247633 0.03737225 0.03448902 0.03950258 0.04441193 0.04149515 0.03366303 0.04054435
 0.03978632 0.04160346 0.0404573  0.03319214 0.03800414 0.03229879 0.04076255 0.03475258
 0.03333832 0.03761944 0.04427279 0.03837532 0.03909892 0.03384874 0.03707403 0.03838878
 0.03682566 0.03278314 0.03412819 0.02993927 0.03138862 0.02646729 0.02687729 0.02550817
 0.02197918 0.01949890 0.02680004 0.02290738 0.01801939 0.01872908 0.01909837 0.01497874
 0.01907834]
```

Notice that the error is about the same at all time steps.
In figure 4 the performance of the trained Neural Network performance on 6 new cases are shown. Each case corresponds to a set of parameters $[r, \kappa, X_0]$ and a new realization of the random noise. The "exact" control u is computed by the brute force method with $K = 100$ and compared with the solution u_{NN} given by the neural network (values at the 50 time steps).
In all cases except case 4 the control u_{NN} given by the network is close to the "exact" one. But even in case 4, X_{NN} computed by (10) with u_{NN} is close to X computed with u "exact".

5. Bellman's Stochastic Dynamic Programming for the dynamic problem

Consider the *value function*

$$V(t,X) = \min_{u\in\mathcal{U}} \int_t^T \mathbb{E}\left[(X_\tau - X_d)^2 - \alpha u + \beta|\partial_t u|^2\right] \mathrm{d}\tau \ : \\ \mathrm{d}X_\tau = X_\tau \mathrm{d}\tau(r - \kappa X_\tau - u_\tau) + \sigma X_\tau \mathrm{d}W_\tau, \quad X(t) = X\} \quad (19)$$

Let $[0,T] = \cup_1^M [(m-1)h, mh]$. Let $\zeta_h(X, u^m, z)$ denote one iterate of a numerical scheme (10)(11) for the SDE starting at X, for instance with (10)

$$\zeta_h(X,u,z) = X + X(r - \kappa X - u)h + X\sigma z\sqrt{h}, \quad z \text{ being the realization of a } \mathcal{N}_{0,1} \text{ r.v.}$$

Let $v^M(X) = 0$ and

$$v^m(X) = \min_{u\in\mathcal{U}} \int_{mh}^{(m+1)h} \mathbb{E}\left[|\zeta_h(X,u,z) - X_d|^2 - \alpha u + \beta|\partial_t u|^2\right] \mathrm{d}\tau + \mathbb{E}[v^{m+1}(\zeta_h(X,u,z)] \\ \approx \min_{u\in\mathcal{U}} \left\{ h\mathbb{E}\left[|\zeta_h(X,u,z) - X_d|^2\right] - \alpha h u + \frac{\beta}{h}|u^{m+1}(X) - u|^2 + \mathbb{E}[v^{m+1}(\zeta_h(X,u,z)] \right\}. \quad (20)$$

Evidently

$$\mathbb{E}\left[|\zeta_h(X,u,z) - X_d|^2\right] = [X - X_d + hX(r - \kappa X - u)]^2 + hX^2\sigma^2.$$

To compute $\mathbb{E}[v^{m+1}(\zeta_h(X,u,z)]$ we use a quadrature formula with Q points $\{z_q\}_1^Q$ and weight $\{\omega_q\}_1^Q$:

$$\mathbb{E}[v^{m+1}(\zeta_h(X,u,z)] \approx \sum_{q=1}^{Q} \omega_q v^{m+1}(\zeta_h(X,u,z_q)$$

Finally at every time step and every $X_j = j\,L/J, j = 1..J$, with $L >> 1$, the result is minimized with respect to $u \in \mathcal{U}$ by a dichotomy. In this fashion $\{u^m(X_j)\}_{j=1}^J$ is obtained and a piecewise linear interpolation is constructed to prepare for the next time step u^{m-1}. With the same parameters as above except $\beta = 0.1$, $L = 3$, $J = 50$ and $M = 50$, the results of figures 5, 6 have been obtained. Once $\{u^m(X)\}_0^{M-1}$ is known, for any realization

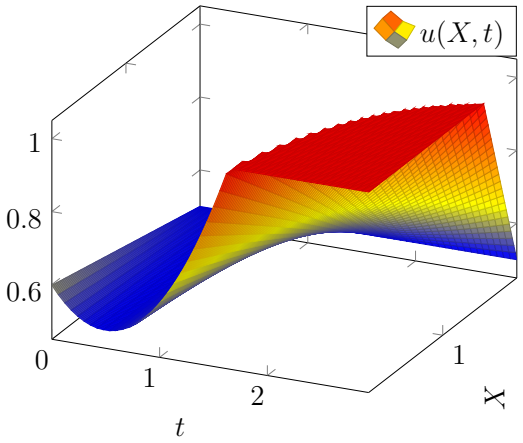

Figure 5: Solution of the dynamic programming equation: $u(x,t)$.

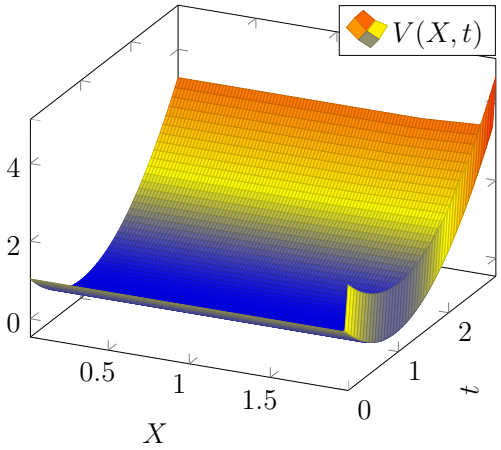

Figure 6: Solution of the dynamic programming equation: $V(x,t)$.

of the fishing model we use it as follows:

$$X^{m+1} = X^m + hX^m(r - \kappa X^m - u^m(X^m)) + X^m\sigma\sqrt{h}\mathcal{N}_{0,1}^m, \quad X^0 = X_0.$$

Two such simulations are shown on figure 7,8. The results are not as good as before, meaning that the dynamic setting gives a poor solution to the control of the fishing site. These results should be compared with those of Appendix B which deals with a very

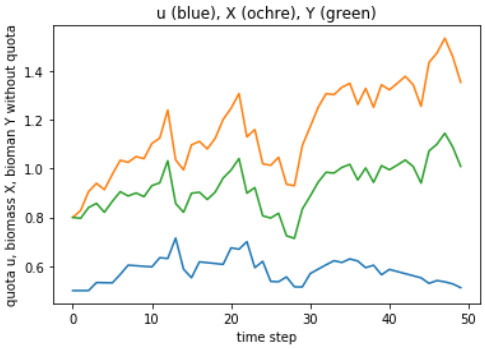

Figure 7: Simulation of the fishing model with a quota function computed by Dynamic Programming and $X_0 = 0.8$.

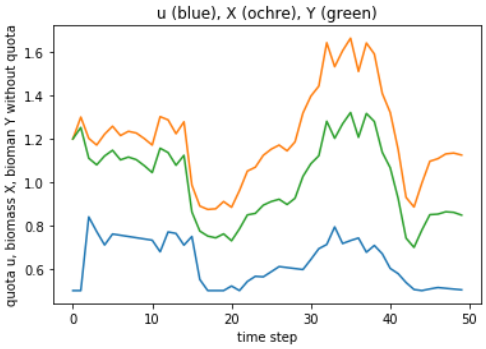

Figure 8: Simulation of the fishing model with a quota function computed by Dynamic Programming and $X_0 = 1.2$.

similar problem for which there is a polynomial in $\log X$ solution when $\alpha = 0$.

6. Solution with the Hamilton-Jacobi-Bellman partial differential equations

The Dynamic Programming equations can be written as a deterministic set of PDE: the Hamilton-Jacobi-Bellman (HJB) equations.
To this end we consider the same value function defined in (19)

Proposition 4. *The value function verifies* $V(\cdot,T)=0$ *and*

$$0=\inf_{u\in\mathcal{U}}\{(X-X_d)^2-\alpha u+\beta|\partial_t u|^2+\partial_t V+(r-\kappa X-u)X\partial_X V+\frac{\sigma^2X^2}{2}\partial_{XX}V\}. \quad (21)$$

Proof
The proof given below is heuristic and non standard but it gives a link with the previous section. For the standard argument see for instance [9]. We return to (20) above:

$$\begin{aligned} v^m(X) &= \min_{u\in\mathcal{U}}\int_{mh}^{(m+1)h}\mathbb{E}\left[|\zeta_h(X,u,z)-X_d|^2-\alpha u+\beta|\partial_t u|^2\right]\mathrm{d}\tau+\mathbb{E}[v^{m+1}(\zeta_h(X,u,z)] \\ &\approx \min_{u\in\mathcal{U}}\left[\mathbb{E}\left[|X-X_d|^2\right]-\alpha u+\frac{\beta}{h^2}|u^{m+1}(X)-u|^2\right]h+\mathbb{E}[v^{m+1}(\zeta_h(X,u,z)] \quad (22)\end{aligned}$$

As u^M plays no role in the discretization we assume that $u^M=u^{M-1}$. Hence when $m=M-1$ the minimization reduces to

$$v^{M-1} = h\min_{u\in\mathcal{U}}\{\mathbb{E}\left[|X-X_d|^2\right]-\alpha u\}=h(|X-X_d|^2-\alpha u_m)$$

and the minimizer is $u^{M-1}=u_m$. When $m<M-1$ a similar argument gives,

$$v^m(X)\approx\min_{u\in\mathcal{U}}\left\{\left(|X-X_d|^2-\alpha u+\frac{\beta}{h^2}|u^{m+1}(X)-u|^2\right)h+\mathbb{E}[v^{m+1}(\zeta_h(X,u,z)]\right\} \quad (23)$$

With a Taylor expansion, let us approximate $v^{m+1}(X+h)\approx\sum_{q=0}^{q_M}{v^{(q)}}^{m+1}(X)\frac{h^q}{q!}$. For clarity denote $\lambda=r-\kappa X-u$, so

$$\mathbb{E}[v^{m+1}(\zeta_h(X,u,z))]=v^{m+1}(X)+\mathbb{E}[\sum_0^{q_M}{v^{(q)}}^{m+1})(X)(X\sqrt{h})^q(\sigma z+\lambda\sqrt{h})^q]\approx v^{m+1}(X)$$

$$+Xv'^{m+1}\lambda h+\frac{X^2}{2}v''^{m+1}(\lambda^2h+\sigma^2)h+\frac{X^3}{6}\lambda\sigma^2{v^{(3)}}^{m+1}h^2+\frac{X^4}{24}\sigma^4{v^{(4)}}^{m+1}h^2+o(h^2)$$

$$\mathbb{E}[v^{m+1}(\zeta_h(X,u,z))]\approx v^{m+1}(X)+Xv'^{m+1}(r-\kappa X-u)h+\frac{X^2}{2}v''^{m+1}\sigma^2h+o(h)$$

Let us use it in (23). Then u must solve

$$\min\left\{-\alpha u+\frac{\beta}{h^2}(u^2-2uu^{m+1})-uXv'^{m+1}\right\},\quad\Rightarrow u=u^{m+1}+\frac{h^2}{2\beta}(\alpha+Xv'^{m+1}) \quad (24)$$

but comply also with the box constraints, i.e.

$$\begin{aligned} u^m &= \min\left\{u_M,\max\left\{u_m,u^{m+1}+\frac{h^2}{2\beta}(\alpha+Xv'^{m+1})\right\}\right\},\ \text{and} \\ v^m &\approx v^{m+1}+h\left(Xv'^{m+1}(r-\kappa X-u^m)+\frac{X^2\sigma^2}{2}v''^{m+1}+|X-X_d|^2-\alpha u^m\right. \\ &\left.+\frac{\beta}{h^2}|u^{m+1}-u^m|^2\right). \quad (25)\end{aligned}$$

As this corresponds to an explicit discretization of a parabolic PDE, h will have to be extremely small: $h=O(\delta X^2$, where δX is the spatial discretization increment.

6.1. Numerical Results

We have implemented the method using `FreeFEM++` [13] with a slight change in (25): the term $\frac{X^2\sigma^2}{2}v''^{m+1}$ is changed to $\frac{X^2\sigma^2}{2}v''^{m}$ so as to increase stability.
Numerical results shown on figure 19, 10 are obtained with the program in file `pagesbellmanstoch.edp` and the following parameters:

$$r = 2, \kappa = 1.2, \sigma = 0.3, X_d = 1, \alpha = 0.01, \beta = 0.1, u_m = 0.5, u_M = 1.$$

The computational domain is $X \in (0, 3)$ divided in 400 intervals, $t \in (0, 2)$, discretized with 40 time steps.

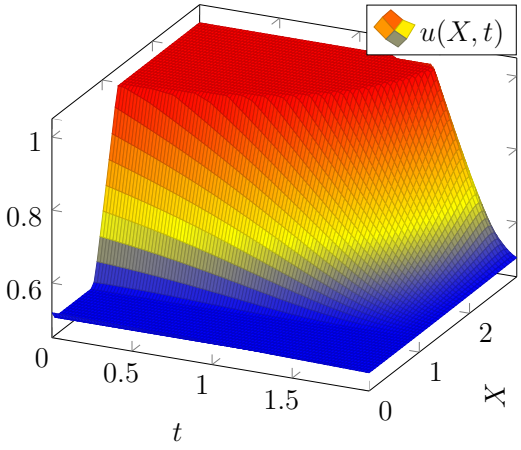

Figure 9: Solution of the dynamic programming equation: side view.

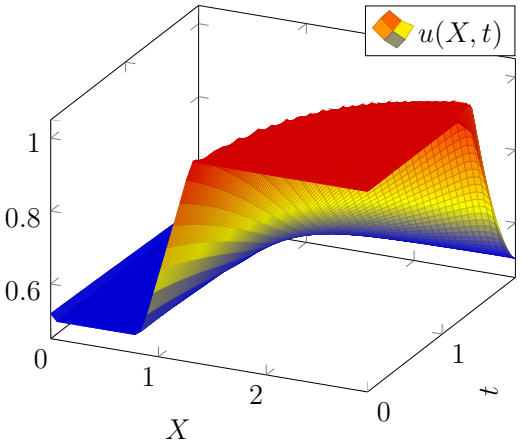

Figure 10: Solution of the dynamic programming equation: front view.

Once $\{u^m(X)\}_0^{M-1}$ is known, for any realization of the fishing model we use it as follows:

$$X^{m+1} = X^m + hX^m(r - \kappa X^m - u^m(X^m)) + X^m\sigma\sqrt{h}\mathcal{N}_{0,1}^m, \quad X^0 = X_0.$$

Two such simulations are shown on figure 11,12.

6.2. Analysis when $\beta = 0$

Proposition 5. *When $\beta = 0$ the solution is always bang-bang.*

Proof: With $\mathcal{U} = \{u \in [u_m, u_M]\}$ and $u_m > 0$, X being positive, the optimal control u^* must be such that if there are $0 < t_1, t_2 < T$ such that for all $t \in (t_1, t_2)$, $u(X, t)$ is constant then

1. $u^* = u_m$ when $X\partial_X V + \alpha < 0$, leading to $\partial_t u^* = 0$ and

$$\partial_t V + (r - \kappa X - u_m)X\partial_X V + \frac{\sigma^2 X^2}{2}\partial_{XX} V = \alpha u_m - (X - X_d)^2$$

2. $u^* = u_M$ when $X\partial_X V + \alpha > 0$, leading to $\partial_t u^* = 0$ and

$$\partial_t V + (r - \kappa X - u_M)X\partial_X V + \frac{\sigma^2 X^2}{2}\partial_{XX} V = \alpha u_M - (X - X_d)^2$$

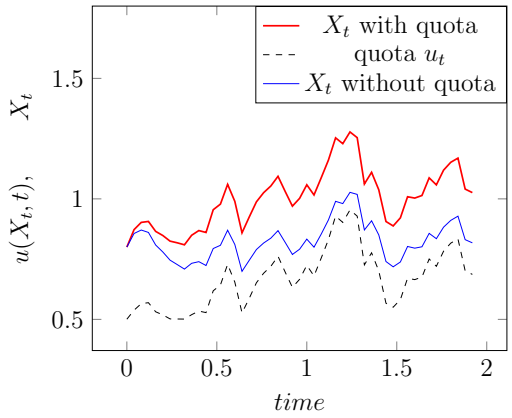

Figure 11: Simulation of the fishing model with a quota function computed by Dynamic Programming and $X_0 = 0.8$.

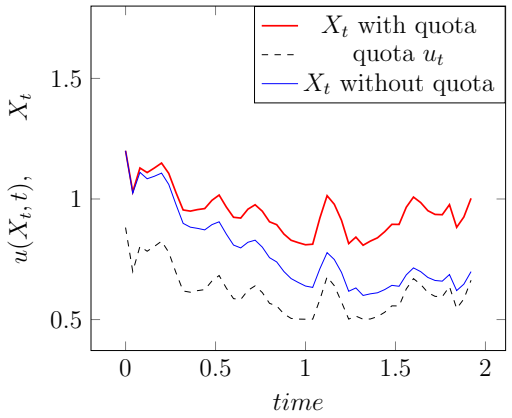

Figure 12: Simulation of the fishing model with a quota function computed by Dynamic Programming and $X_0 = 1.2$.

3. Otherwise when $u_m < u^*(X,t) < u_M$, we must have $V(T) = 0$ and

$$0 = (X - X_d)^2 + \partial_t V + (r - \kappa X) X \partial_X V + \frac{\sigma^2 X^2}{2} \partial_{XX} V, \quad X \partial_X V + \alpha = 0$$

The third option being impossible, the solution of the problem is bang-bang.

6.3. Discretization

One way to find u^* when it is neither u_m nor u_M is to discretize in time and solve

$$\inf_{u^m} g(u^m) := \{-\alpha u^m + \frac{\beta}{h^2} |u^m - u^{m+1}|^2 - u^m X \partial_X V\}$$

The stationary point is at

$$-\alpha + \frac{2\beta}{h^2}(u^{m-1} - u^m) - X \partial_X V^{m-1} = 0, \text{ i.e. } \bar{u}^{m-1} = u^m + \frac{h^2}{2\beta}(\alpha + X \partial_X V^{m-1}).$$

So we have to compare $g(u_m), g(\bar{u}^m)$ and $g(u_M)$ and choose the smallest. The end result is very similar to the previous section.
Two simulations are done with $\beta = 0$ and shown on figure 13,14. The control is indeed bang-bang.

7. Solution with the Kolmogorov equation

The Kolmogorov equation for the probability density $\rho(X,t)$ of $\{X_t\}_0^T$ is:

$$\partial_t \rho + \partial_X[(r - \kappa X - u) X \rho] - \partial_{XX}[\frac{X^2 \sigma^2}{2} \rho] = 0, \ \rho(X,0) = \rho^0(X), \ \forall X \in \mathbb{R}^+. \qquad (26)$$

The optimal control problem (1) is:

$$\min_{u \in \mathcal{U}} J(u) := \int_{\mathcal{Q}^+} \left[(X - X_d)^2 - \alpha u(X,t) + \beta |\partial_t u(X,t)|^2\right] \rho(X,t) \mathrm{d}X \mathrm{d}t \ : \text{ with (26). (27)}$$

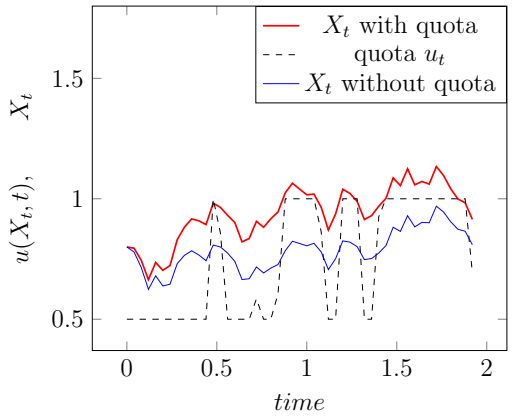

Figure 13: Simulation of the fishing model with a quota function computed by HJB when $\beta = 0$ and $X_0 = 0.8$.

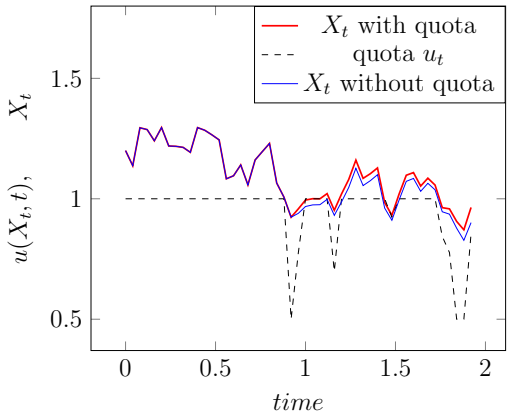

Figure 14: Simulation of the fishing model with a quota function computed by HJB when $\beta = 0$ and $X_0 = 1.2$.

Integration is done on $\mathcal{Q}^+ := \mathbb{R}^+ \times]0, T[$. The conditions for having equivalence between the stochastic control written as (1) and (27) are detailed in [11]. To avoid using weighted Sobolev spaces we make a change variable $y = \log X$, set $v = r - \frac{\sigma^2}{2} - \kappa e^y - u(e^y, t)$ so that $\mathrm{d}Y_t = v\mathrm{d}t + \sigma \mathrm{d}W_t$ and then study the existence of a PDF for Y_t given by

$$\partial_t \rho + \partial_Y (v\rho) - \partial_{YY}[\frac{\sigma^2}{2}\rho] = 0, \ \rho(Y,0) = \rho^0(Y), \ \forall Y \in \mathbb{R}, \ \forall t \in]0,T[. \tag{28}$$

The PDE is understood in the sense that for all $\hat{\rho} \in H^1(\mathbb{R})$,

$$\int_{\mathbb{R}} \left[\hat{\rho}\partial_t \rho - v\rho \partial_Y \hat{\rho} + \frac{\sigma^2}{2} \partial_Y \rho \partial_Y \hat{\rho} \right] = 0, \ a.e. \ t \in]0,T[; \ \rho(Y,0) = \rho^0(Y), \ \forall Y \in \mathbb{R}. \tag{29}$$

Note that v can be changed to $v - L$, $L \in \mathbb{R}$ by changing ρ to ρe^{-Lt}. Hence when $\lim_{Y\to\infty} u = u_\infty \neq 0$ we can take $L = r - \frac{\sigma^2}{2} - u_\infty$ and assume that $\lim_{Y\to\infty} v = 0$.

Proposition 6. *If $v \in W^{1,1}_{loc}(\mathbb{R})$, $v/(1+|y|) \in L^1(\mathbb{R}) \cap L^\infty(\mathbb{R})$, $\partial_y v \in L^\infty(\mathbb{R})$ then Y_t has a PDF, $\rho \in L^\infty\left(L^2(\mathbb{R}) \cap L^\infty(\mathbb{R})\right) \cap L^2\left(H^1(\mathbb{R})\right)$, given by (28).*

Consequently, for the problem to make sense, $(r - \kappa X - u)$ needs to be multiplied by a regular cutoff function, tending to zero fast enough when $X \to +\infty$. We shall henceforth assume that κ is a function of X which is zero when $X > X_M$, for some X_M.

Theorem 2. *Given $X_d, \sigma, \alpha, \beta, K, T, R$ in $\mathbb{R}^+$, v_m, v_M, K in $\mathbb{R}$ and $\rho^0 \in [0,1]$ with $\int_{\mathbb{R}} \rho^0 = 1$, let $\mathcal{Q} = \mathbb{R}\times]0,T[$ and $\mathcal{Q}_R =]-R, R[\times]0,T[$ and,*

$$\mathcal{V} = \{v : \ v_m \leq v \leq v_M, \ \|\partial_t v\|_{L^2(\mathcal{Q})} \leq K\}$$

and consider

$$\min_{v\in\mathcal{V}} J(v) := \int_{\mathcal{Q}_R} \left[(e^Y - X_d)^2 + \alpha v(Y,t) + \beta |\partial_t v(Y,t)|^2\right] \rho(Y,t) \mathrm{d}Y \mathrm{d}t \ : \ with \tag{30}$$

$$\partial_t \rho + \partial_Y (v\rho) - \partial_{YY}[\frac{\sigma^2}{2}\rho] = 0, \ \rho(Y,0) = \rho^0(Y), \ \forall Y \in \mathbb{R}, \ \forall t \in]0,T[. \tag{31}$$

Problem (30) has a solution.

Proof: Note first that $J \geq \alpha v_m$. Note also that when u_M is large enough so that $v = 0 \in \mathcal{V}$, the solution of the PDE is $\tilde{\rho}(Y,t) = (4\pi)^{-\frac{1}{2}} \int_{\mathbb{R}} e^{-\frac{X^2}{4t}} \rho^0(X-Y)\mathrm{d}X$. As $\int_{-R}^{R} \tilde{\rho}(Y,t)e^{2Y}$ is finite and $\mathcal{V}$ is non empty the infimum is finite. Multiplying the PDE by ρ and integrating over $\mathbb{R}$ leads to

$$\int_{\mathbb{R}} [\partial_t |\rho^2| + \sigma^2 |\partial_Y u|^2] \leq 2 \int_R |v| \rho \partial_Y \rho \leq 2 \|v\|_\infty \|\rho\|_{L^2(\mathbb{R})} \|\partial_Y \rho\|_{L^2(\mathbb{R})}, \quad a.e.\ t \in]0,T[$$

Consequently, and because $\|v\|_\infty \leq v_M$, ρ is bounded in $L^2\left(H^1(\mathbb{R})\right) \cap L^\infty\left(L^2(\mathbb{R})\right) \cap L^2(\mathcal{Q})$ and $\partial_t \rho$ is bounded in $L^2\left(H^{-1}(\mathbb{R})\right)$. Consequently ρ is compact in $L^2(\mathcal{Q})$ when v varies in $\mathcal{V}$. From the maximum principle we also know that $0 \leq \rho_n \leq 1$ in $\mathcal{Q}$ and that $\int_{\mathbb{R}} \rho(Y,t)\mathrm{d}Y = 1$ a.e.
Now let us take a minimizing sequence $\{v_n\}_{n\in\mathbb{N}}$ for (30). Then ρ_n is bounded in $L^2\left(H^2(\mathbb{R})\right) \cap L^\infty\left(L^2(\mathbb{R})\right) \cap W^{1,2}\left(H^{-1}(\mathbb{R})\right)$ and compact in $L^2(\mathcal{Q}_R)$. It follows from above that there is a v such that, for a subsequence, $v_n \to v$ inL^∞-weak-star and weakly in $L^2(\mathcal{Q})$ and a ρ such that $\rho_n \to \rho$ strongly in $L^2(\mathcal{Q})$. Hence $\int_{\mathbb{R}} v_n \rho_n \partial_Y \hat{\rho} \to \int_{\mathbb{R}} v \rho \partial_Y \hat{\rho}$ for any $\hat{\rho} \in L^2\left(H^1(\mathbb{R})\right)$ and consequently ρ solves the PDE with v .
As $\rho_n \in [0,1]$ and by Fatou's lemma, $\int_{\mathcal{Q}_R} (e^Y - X_d)^2 \geq \liminf \int_{\mathcal{Q}_R} (e^Y - X_d)^2 \rho_n \geq \int_{\mathcal{Q}} (e^Y - X_d)^2 \rho$.
By the weak convergence of v_n and the strong convergence of ρ_n, $\int_{\mathcal{Q}_R} \alpha v_n \rho_n \to \int_{\mathcal{Q}_R} \alpha v \rho$.
To analyze the last term in the cost function we write it as

$$\beta \int_{\mathcal{Q}_R} |\partial_t v_n|^2 \rho_n = \beta \int_{\mathcal{Q}_R} |\partial_t v_n|^2 \rho + \beta \int_{\mathcal{Q}_R} |\partial_t v_n|^2 (\rho_n - \rho).$$

By the lower semi-continuity of semi-norms, $\lim_{n\to\infty} \int_{\mathcal{Q}_R} |\partial_t v_n|^2 \rho \geq \int_{\mathcal{Q}_R} |\partial_t v|^2 \rho$. Finally the last term above is bounded by $\beta \|(\partial_t v_n)^2\|_{L^2(\mathcal{Q}_R)} \|\rho_n - \rho\|_{L^2(\mathcal{Q}_R)} \leq \beta K \|\rho_n - \rho\|_{L^2(\mathcal{Q}_R)} \to 0$. Consequently, the cost function in (30) is l.s.c.

Remark 1. *If $\int_{\mathbb{R}}^{+} X^2 \rho^0(X)\mathrm{d}X < \infty$, it may be possible to let $R \to \infty$ in the proof above.*

7.1. Computation of gradients
Consider the variational form of the Kolmogorov equation: find $\rho \in L^2(0,T,V)$ such that, for all $\hat{\rho} \in V$,

$$\int_{\mathbb{R}^+} \left(\hat{\rho}\partial_t \rho - (r - \kappa X - u) X \rho \partial_X \hat{\rho} + \frac{\sigma^2}{2} \partial_X (X^2 \rho) \partial_X \hat{\rho} \right) = 0, \quad \rho(0) \text{ given.} \tag{32}$$

Calculus of variations leads to $\delta\rho(0) = 0$ and

$$\int_{\mathbb{R}^+} \left(\hat{\rho}\partial_t \delta\rho - (r - \kappa X - u) X \delta\rho \partial_X \hat{\rho} + \frac{\sigma^2}{2} \partial_X (X^2 \delta\rho) \partial_X \hat{\rho} \right) = - \int_{\mathbb{R}^+} \rho X \partial_X \hat{\rho} \delta u. \tag{33}$$

Define the adjoint p' by $p'(T) = 0$ and, for all $\hat{p} \in V$:

$$\int_{\mathbb{R}^+} \Big(\hat{p}\partial_t p' + \hat{p}(r - \kappa X - u) X \partial_X p' \quad - \frac{\sigma^2}{2} \partial_X (X^2 \hat{p}) \partial_X p' \\ + \hat{p} \mathbf{1}_\Omega [(X - X_d)^2 - \alpha u + \beta |\partial_t u|^2] \Big) = 0. \tag{34}$$

Add (33) and (34) with $\hat{\rho} = p'$ and $\hat{p} = \delta\rho$. It gives

$$\int_{\mathbb{R}^+} \partial_t(p'\delta\rho) + \int_\Omega [(X - X_d)^2 - \alpha u + \beta|\partial_t u|^2]\delta\rho = \int_{\mathbb{R}^+} \rho X \partial_X p' \delta u.$$

As $p'(T) = 0$ and $\delta\rho(0) = 0$, an integration in time gives

$$\int_{\Omega\times]0,T[} [(X - X_d)^2 - \alpha u + \beta|\partial_t u|^2]\delta\rho = -\int_{\mathcal{Q}^+} \rho X \partial_X p' \delta u.$$

Finally, from (27),

$$\begin{aligned} \delta J &= < grad_u J, \delta u >= \int_{\Omega\times[0,T]} \left[\left((X - X_d)^2 - \alpha u + \beta|\partial_t u|^2\right)\delta\rho - \alpha\rho\delta u + 2\rho\partial_t u\partial_t\delta u\right]. \\ &= -\int_{\mathcal{Q}^+} \rho X \partial_X p' \delta u + \int_{\Omega\times]0,T[} \rho[-\alpha\delta u + 2\partial_t u\partial_t\delta u] \end{aligned} \tag{35}$$

7.2. Results

A steepest descent algorithm with fixed step size and projected gradient on the box constraints is

$$v = u^k - \mu grad_u J^k, \quad u^{k+1} = \max\{u_m, \min\{u_M, v\}\}. \tag{36}$$

Note that we have assumed that $\partial_t u(0) = \partial_t u(T) = 0$. The same parameters as above have been used except that the computational domain is $(0, 5)$ instead of $(0, 3)$. The PDF of X_0 is a Gaussian curve with $\sigma_0 = 0.2$. The computer program is in the file `GPOP2dxt.edp`. After 100 iterations J is a fifth of its initial value. Figure 15 shows the surfaces u,P and ρ versus X, t. Note that $(X, t) \mapsto u(X, t)$ is either 0.5 or 1, implying that all controls $u(X_t, t)$ will be bang-bang. Figures 16 and 17 show two applications, one with $X_0 = 0.8$ and the other with $X_0 = 1.2$. The method achieves a reasonable stabilization of the fish biomass, but the quotas vary rapidly, as if the β -term wasn't playing its regularizing role.

8. Solution by Itô calculus

In computational finance Itô calculus is preferred over the Kolmogorov setting, primarily because the initial condition for the PDF could be singular.
Consider again Problem (1). Itô calculus applied to

$$j_t = \int_t^T \mathbb{E}\left[(X_\tau - X_d)^2 + \beta(\partial_t u)^2 - \alpha u\right] \mathrm{d}\tau$$

as a function of X_t gives:

$$\mathrm{d}_t j = \partial_t j \mathrm{d}t + \mathbb{E}[\partial_X j \mathrm{d}X] + \frac{1}{2}\mathbb{E}[\partial_{XX} j \mathrm{d}X \mathrm{d}X]$$

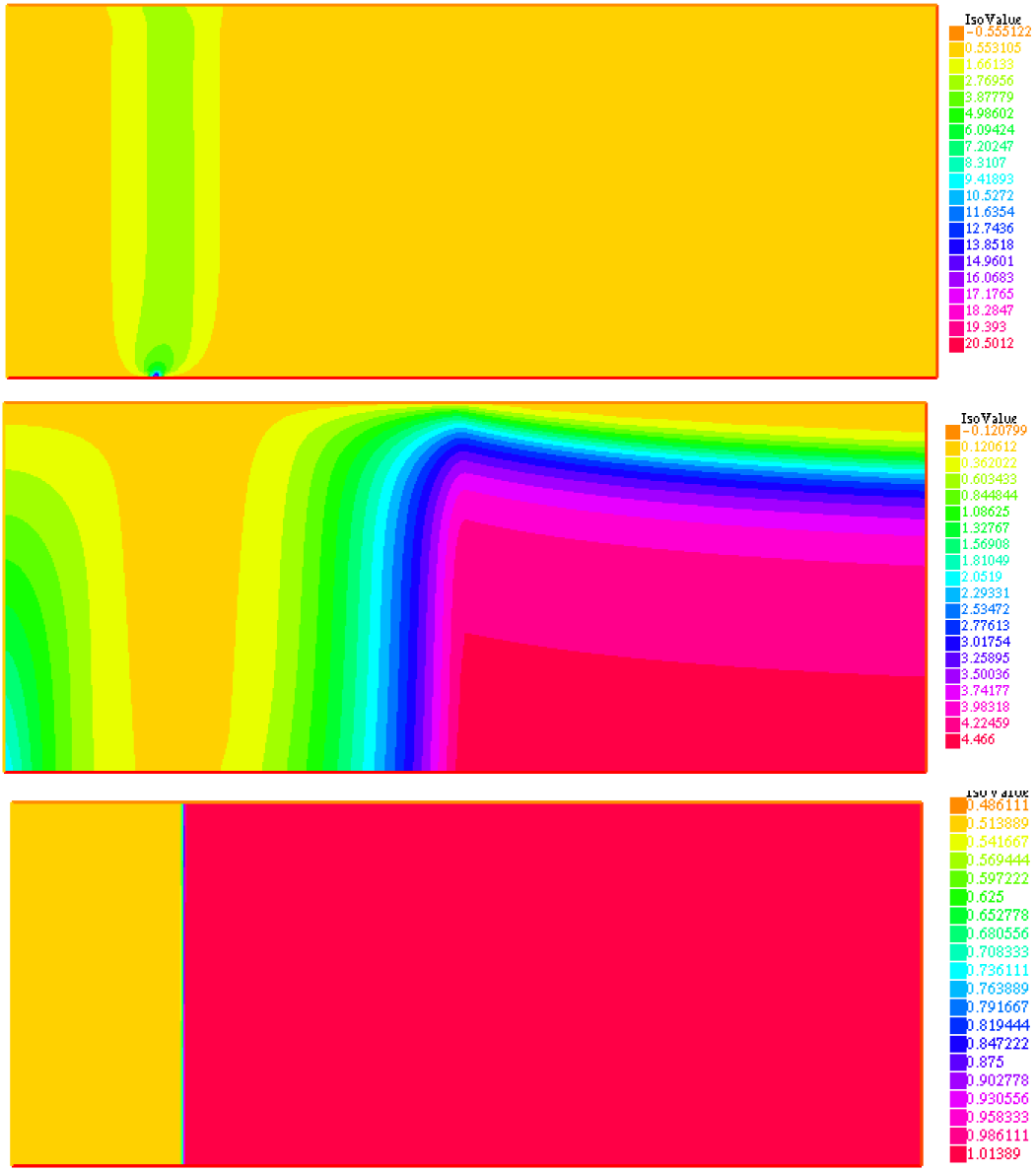

Figure 15: From left to right: optimal ρ,p',u. Horizontal axis is $X \in]0, X_M[$, vertical is $t \in [0, T]$ with $X_M = 5$, $T = 2$, $X_d = 1$, $u_m = 0.5$, $u_M = 1$. Note that $(X, t) \mapsto u(X, t)$ is either 0.5 or 1, implying that all controls $u(X_t, t)$ will be bang-bang.

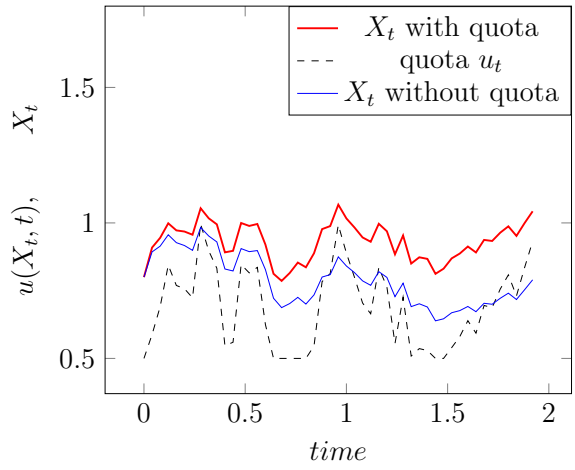

Figure 16: Optimal biomass and quota function computed by Optimal Control and compared with the solution with $u = u_M$ when $X_0 = 0.8$.

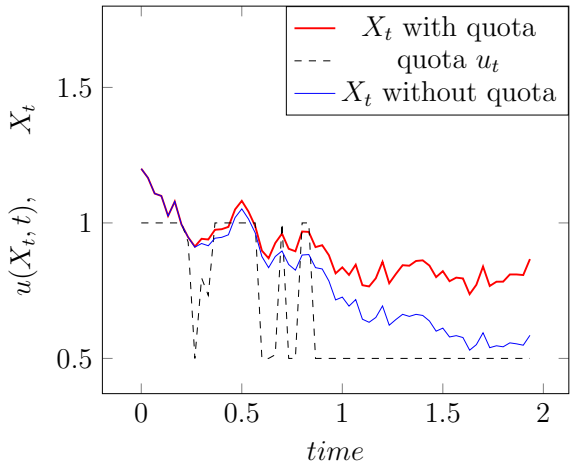

Figure 17: Optimal biomass and quota function computed by Optimal Control and compared with the solution with $u = u_M$ when $X_0 = 1.2$.

$$= \mathrm{d}t\left(\partial_t j + \partial_X j(r - \kappa X - u)X + \frac{X^2\sigma^2}{2}\partial_{XX} j\right) = -\mathrm{d}t[(X - X_d)^2 + \beta(\partial_t u)^2 - \alpha u] \tag{37}$$

This formulation is dual to the Kolmogorov calculus presented above.
Let us compute j as the solution at $\tau = 0$ and $X = X^0$ of the PDE, backward in time and initiated by $j(X,T) = 0,\ \forall X \in \mathbb{R}^+,\ \tau \in]0,T[$:

$$\partial_t j + (r - \kappa X - u)X\partial_X j + \frac{X^2\sigma^2}{2}\partial_{XX} j = -[(X - X_d)^2 + \beta(\partial_t u)^2 - \alpha u] \tag{38}$$

The optimal control problem (1) is now

$$\min_{u\in\mathcal{U}} \int_{\mathbb{R}} j(X,0)\rho^0(X)\mathrm{d}X \ :\ \text{subject to (38)} \tag{39}$$

where $\rho^0(X)$ is the PDF of X^0. We have to restrict $X < X_M$ for the right hand side of the PDE to be in $L^2(\mathbb{R})$.
On $\Omega := (0, X_M)$ the variational formulation of the PDE consists in finding j in a weighted Sobolev space:

$$\begin{aligned} &j \in V := \{g \in L^2(\Omega)\ :\ X\partial_X g \in L^2(\Omega)\},\quad j(\cdot,T) = 0,\\ &\int_\Omega (\partial_t j + (r - \kappa X - u)X\partial_X j)\,\hat{j} - \int_\Omega \frac{\sigma^2}{2}\partial_X(X^2\hat{j})\partial_X j + \frac{\sigma^2}{2}X_M j(X_M)\hat{j}(X_M)\\ &= -\int_\Omega [(X - X_d)^2 + \beta(\partial_t u)^2 - \alpha u]\hat{j},\ \forall \hat{j} \in V. \end{aligned} \tag{40}$$

8.1. Gradient computation

The gradient of j with respect to u results from the following calculus:

$$\int_{\Omega\times]0,T[} grad_u j\delta u := \int_\Omega \rho^0\delta j(X^0,0) \text{ subject to}$$

$$\int_\Omega (\partial_t \delta j + (r - \kappa X - u) X \partial_X \delta j)\, \hat{j} - \int_\Omega \frac{\sigma^2}{2} \partial_X (X^2 \hat{j}) \partial_X \delta j$$
$$+ \frac{\sigma^2}{2} X_M \delta j(X_M) \hat{j}(X_M) = \int_\Omega \hat{j}[(X \partial_X j + \alpha)\delta u - 2\beta \partial_t u \partial_t \delta u],\ \forall \hat{j} \in V. \tag{41}$$

Define p by

$$\int_\Omega (\hat{p}\partial_t p - (p(r - \kappa X - u) X \partial_X \hat{p}) + \int_\Omega \frac{\sigma^2}{2} \partial_X (X^2 p) \partial_X \hat{p} - \frac{\sigma^2}{2} X_M \hat{p}(X_M) p(X_M) = 0,$$
$$p(0) = \rho^0 \tag{42}$$

Then adding (42) to (41) with $\hat{p} = \delta j$ and $\hat{j} = p$ and remembering that $\delta j(X, T) = 0$, leads to

$$\int_\Omega (p \partial_t \delta j + \delta j \partial_t p) = \int_{\Omega \times]0,T[} [p(X \partial_X j + \alpha)\delta u - 2\beta \partial_t u \partial_t \delta u] \tag{43}$$

On the other hand

$$-\int_\Omega \rho^0 \delta j(X, 0) = \int_\Omega \delta j(X, T) p(T) - \int_\Omega \delta j(X, 0) p(0) = \int_{\Omega \times]0,T[} \partial_t (p \delta j) \tag{44}$$

Consequently

$$\int_{\Omega \times]0,T[} grad_u j \delta u = \int_\Omega \rho^0 \delta j(X^0, 0) = -\int_{\Omega \times]0,T[} [p(X \partial_X j + \alpha)\delta u - 2\beta \partial_t u \partial_t \delta u]$$

Hence this method is a minor variation of the method with the Kolmogorov equation: ρ and p have similar equations and p' and j too.

9. Limit with vanishing volatility

We have seen earlier that the optimal solution of the non-dynamic control is very close to the solution of the deterministic control problem.
We explore here the convergence of the PDE method to the deterministic case by letting σ tend to zero. For clarity we take $\alpha = \beta = 0$.
Consider again (39). Let $Y(X, \tau)$ be the solution of

$$\dot{Y}(\tau) = (r - \kappa Y(\tau) - u(Y(\tau), \tau))\, Y(\tau), \quad Y(t) = X; \tag{45}$$

then

$$\partial_t j + (r - \kappa X - u) X \partial_X j = \frac{\mathrm{d}}{\mathrm{d}\tau} \rho(Y(\tau, X), \tau)|_{\tau = t}. \tag{46}$$

Hence (38) is rewritten as

$$\frac{\mathrm{d}}{\mathrm{d}\tau} j(Y(\tau, X), \tau)|_{\tau = t} = -\frac{\sigma^2}{2} \partial_{XX}(X^2 \rho) - (X - X_d)^2.$$

Consequently, when $\sigma \to 0$, j becomes equal to $\tilde{j}$, the integral of $-(X - X_d)^2$ on the streamline (45) which passes through X^0 at $t = 0$:

$$\tilde{j}(Y(T), T) - \tilde{j}(Y(0), 0) = -j(X^0, 0) = -\int_{Y(0)}^{Y(T)} (X(s) - X_d)^2$$
$$i.e.\ \tilde{j}(0) = \int_0^T (X(t) - X_d)^2 \mathrm{d}t\ :\ \dot{X} = (r - \kappa X - u)X, \quad X(0) = X^0. \tag{47}$$

10. Concluding discussion

Let us compare the 3 classes of methods:

1. Non-dynamic control (NDC)
2. Dynamic Programming (DP)
3. Dynamic control solved with Partial Differential equations (HJB).

NDC is a mathematical oddity because it does not take into account the necessity of a Markovian solution and it does not attempt to use the current noise level to predict the next step, but from the practical point of view it is certainly the best solution for the fishing site regulator, especially when it is coupled with Statistical Learning. After all, provided that the model fits the reality, a Markovian solution is not needed.
Note that there is large literature on Model Predictive Control (see [7]) to use the current state to predict the next step without using dynamic programming. Here it can be seen on figure 4 that the precision is hardly better at early times than at later times; hence breaking the control interval $]0, T[$ into smaller intervals (T_j, T_{j+1}) to use the current state as initial condition, is easy to do but probably not worth the effort.
On this fishery problem DP and HJB are different implementations of the same approach. All four implementations gave disappointing results, probably because the parameter chosen gave a solution close to a bang-bang saturation of constraints and not far from the elementary but intuitive strategy which takes $u_t = u_m$ when $X_t > X_d$ and u_M otherwise. But it was said earlier that this is not feasible as it would be unpopular to change too often from a constraining quota to a loose one. HJB with PDEs is the most taxing method numerically and DP is much faster and also capable of generating approximate solution which do not saturate the constraints as much. But still! brute force Monte-Carlo with neural network training for NDC outperform traditional HJB and DP methods.
This problem belongs to a class for which the user cannot measure precisely the random errors in the data and the model; the number of boats at sea and the quantity of fish caught everyday may just reflect the roughness of the model rather than the stochasticity of the measures; nevertheless this point needs to be investigated further and in the mean time the non-dynamic control problem makes sense and gives better results. Furthermore, when coupled with a neural network the optimization process is very fast, once the learning phase is done.

Acknowledgements

Our sincere gratitude goes to François Murat for his help to minimize the hypotheses necessary for the existence of a solution to the Kolmogorov distributed control problem. The computer programs in Python and FreeFEM++ can be obtained by email from `olivier.pironneau@sorbonne-university.fr`.

References

[1] R. Bellman, *Dynamic Programming*, Princeton, NJ: Princeton University Press, 1957.

[2] S.Balakrishnan and V. Biega. Adaptive-critic-based neural networks for aircraft optimal control. Journal of Guidance, Control and Dynamics, 19(4), 893–898. 1996.

[3] G. Pages and Huyên Pham and J. Printems: An Optimal Markovian Quantization Algorithm For Multi-Dimensional Stochastic Control Problems. Stochastics and Dynamics Vol. 04, No. 04, pp. 501-545, 2004.

[4] D. Bertsekas, *Reinforced Learning and Optimal Control*. Athena Scientific, Belmont Mass. 2019.

[5] R. Kamalapurkar and P. Walters and J. Rosenfeld and W. Dixon, *Reinforcement Learning for Optimal Feedback Control*. Springer 2018.

[6] . P. Auger and O. Pironneau, Parameter Identification by Statistical Learning of a Stochastic Dynamical System Modelling a Fishery with Price Variation. Comptes-Rendus de l'Académie des Sciences. May 2020.

[7] S. Boyd and C. Barratt: Linear Controller Design – Limits of Performance. Prentice-Hall, 1991.

[8] T. Brochier, P. Auger, D. Thiao, A. Bah, S. Ly, T. Nguyen Huu, P. Brehmer. Can overexploited fisheries recover by self-organization? Reallocation of the fishing effort as an emergent form of governance. Marine Policy, 95 (2018) 46-56.

[9] Jiongmin Yong and Xun Yu Zhou: *Stochastic Controls Hamiltonian Systems and HJB Equations* Application of Mathematics series vol 43. Springer 1991.

[10] F. Chollet : *Deep learning with Python*. Manning publications (2017).

[11] Le Bris, C., Lions P.L, Existence and uniqueness of solutions to Fokker-Planck type equations with irregular coefficients. Comm. Partial Differential Equations, 33, 1272-1317 , 2008.

[12] I. Goodfellow, Y. Bengio and A. Courville (2016): *Deep Learning*, MIT-Bradford.

[13] F. Hecht (2012): New development in FreeFem++, J. Numer. Math., 20, pp. 251-265. (see also www.freefem.org.)

[14] M. Lauriere and O. Pironneau : Dynamic programming for mean-field type control J. Optim. Theory Appl. 169 (2016), no. 3, 902–924.

APPENDIX A

11. A model with fishing quota

Let $X(t)$ be the fish biomass at time t, $E(t)$ the fishing effort – interpreted as the number of boats at sea – and $Q(t)$ is the fishing capacity per boat. In [8] $Q(t) = qX(t)$,with the catchability q constant, meaning that the more fish there is the more fishermen will catch them.
In this study a quota $Q(t)$, given to each fisherman, is imposed on the maximum weight of fish caught on a day t ; hence the total amount of fish caught on a day is $\min(qX(t), Q(t))E(t)$. The modified model is an equation for $X(t)$ which says that the biomass changes due to the natural growth/decay rate r, the long time limit κ' of X and the depletion due to fishing:

$$\frac{\mathrm{d}X}{\mathrm{d}t}(t) = X(t)(r - \kappa' X(t)) - \min(qX(t), Q(t))E(t). \qquad \text{Biomass dynamics.} \quad (48)$$

Let $F(t) = \min(qX(t), Q(t))$. The fishing effort is driven by the profit pF minus the operating cost of a boat c, where p being the price of fish.

$$\frac{1}{E}\frac{\mathrm{d}E}{\mathrm{d}t} = pF - c. \qquad \text{Fishing effort driven by profit,} \quad (49)$$

The price is driven by the difference between the demand $D(p)$ and the resource FE:

$$\Phi\frac{\mathrm{d}p}{\mathrm{d}t} = D(p) - FE \quad \text{with the demand } D(p) = \frac{a'}{1+\gamma p} \quad (50)$$

where Φ is the inverse time scale at which the fish market price adjusts. When $\Phi << 1$, (50) may be approximated by

$$D(p) = FE \;\Rightarrow\; pFE = \frac{a'}{\gamma} - \frac{FE}{\gamma}.$$

With $a = a'/\gamma$ and $\kappa = \kappa'\gamma$, the whole system is :

$$\frac{\mathrm{d}X}{\mathrm{d}t} = X(r - \frac{\kappa}{\gamma}X) - \min(qX(t), Q(t))E, \quad \frac{\mathrm{d}E}{\mathrm{d}t} = a - \min(qX(t), Q(t))\frac{E}{\gamma} - cE, \quad (51)$$

Now notice that by changing the initial conditions $X_0 \to X_0/\gamma$ we may take $\gamma = 1$.

Remark 2. *Finally with $\tilde{t} = tq$, $\tilde{Q} = Q/q$ and $(\tilde{r}, \tilde{\kappa}", \tilde{a}, \tilde{c}) = (r, \kappa, a, c)/q$ the above system is identical but now $q = 1$.*

Remark 3. *The ODE system is of the form $\dot{Y}(t) = \psi(Y(t)),\ t \in]0, T[,\ \psi : \mathbb{R}^2 \to \mathbb{R}^2$. Global existence of solution on $[0, T]$ is not known because ψ is only locally Lipschitz and not sublinear for large E, X.*

A numerical simulation with and without optimal quota is shown on figure 18 with the following parameters (see file `fishingQuota0.py` and for the definition of α and β, see below):

$$\begin{aligned} &r = 2, \kappa = 1.2, a = 1, c = 1, T = 2, M = 50, X_d = 1, q = 1.5 \\ &X_0 = 0.9, E_0 = 2, \alpha = 0.01, \beta = 0.001, u_m = 0.5, u_M = 1.5 \end{aligned} \quad (52)$$

Notice that q is big enough so that $\min(qX(t), Q(t)) = Q(t),\ \forall t$.

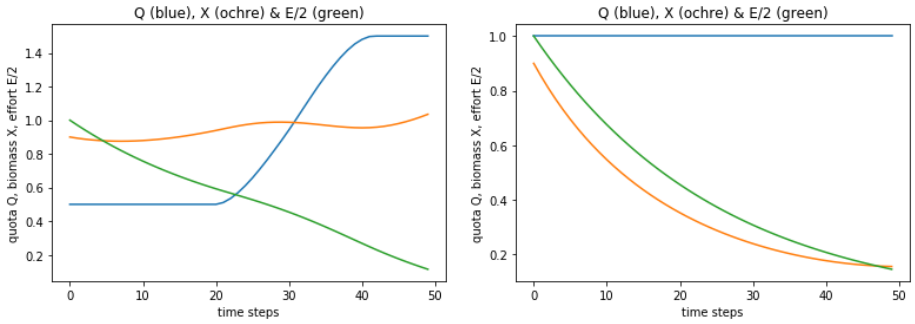

Figure 18: Left: Quota (blue) and fishing effort (ochre). No quota means that $Q(t) = 1$ for all t. Right: biomass with (blue) and without quota (ochre).

12. Reformulation

To prevent fish extinction a constraint is set on the total catch $\min(qX(t), Q(t))E(t))$. Optimization will then define $Q(t)$. It is expected that $Q(t) < qX(t)$ otherwise the policy of quota is irrelevant in the sense that the fisherman is a maximum allowed catch which is greater than what he could possibly catch.
So let us add a constraint $Q(t) \leq Q_M$ so that $\min(qX(t), Q(t)) = Q(t)$. Denote $u(t) = E(t)Q(t)/X(t)$. Then we need $u(t) \leq qE(t)$ and the problem becomes

$$\min_{Q' \in \mathcal{Q}'} \{\bar{J} := \int_0^T (X(t) - X_d(t))^2 \mathrm{d}t \ : \ \frac{\mathrm{d}X}{\mathrm{d}t} = X\left(r - \kappa' X - u(t)\right), \quad X(0) = X^0\} \tag{53}$$

Here the optimization is performed without computing E; then E is computed with the optimal u by solving

$$\frac{\mathrm{d}E}{\mathrm{d}t} = a - cE - uX, \quad E(0) = E^0.$$

As said before it is not feasible to impose to small a quota, so finally $\mathcal{U} = \{u_m \leq u(t) \leq u_M\}$. where u_M is less than $qE(t)$ at all time.

12.1. More constraints on quotas

Small quotas are unpopular. To avoid them we may use penalty and add to the minimization $-\int_0^T \alpha u(t)$.
More over, it will be unpopular if $u(t) = u_m$ one day and $u(t) = u_M$ the next day, so to avoid too many changes we may add to the minimization a term like $\beta|\frac{du}{dt}|^2$.
An optimal policy is a solution of

$$\begin{aligned} \min_{Q \in \mathcal{Q}} J(u) \ &:= \int_0^T \left[|X - X_d|^2 - \alpha u + \beta|\frac{du}{dt}|^2\right] \ : \ \text{subject to} \\ &\frac{\mathrm{d}X}{\mathrm{d}t} = X\left(r - \kappa' X - u(t)\right), \quad X(0) = X^0. \end{aligned} \tag{54}$$

Figure 1 shows the solution with the same parameters as for figure 18 There are differences because it is a different set of constraints. The policy given to each fisherman is $Q(t)$ for the full system and $u(t)X(t)/E(t)$ for the reduced system.

APPENDIX B

13. An analytical solution for a similar problem

It was said earlier that $Y_t = \log X_t$ could be used to simplify the problem. As the goal is to reach $X_t \sim X_d$, if we change it to $Y_t \sim Y_d$, then the problem is

$$\min_{u\in\mathcal{U}}\{\int_0^T \mathbb{E}[(Y_t - Y_d)^2 - \alpha u + \beta|\partial_t u|^2 \text{ with } \mathrm{d}Y_t = v_t\mathrm{d}t + \sigma\mathrm{d}W_t,\ v_t = r - \frac{\sigma^2}{2} - \kappa e^{Y_t} - u\}.$$
$$\Rightarrow\ 0 = \inf_{u\in\mathcal{U}}\{\partial_t V + v\partial_Y V + \frac{\sigma^2}{2}\partial_{YY}V + (Y - Y_d)^2 - \alpha u + \beta|\partial_t u|^2\} \tag{55}$$

If the constraints are not active, the equation for the optimality of u is

$$\partial_Y V + \alpha + 2\beta\partial_{tt}u = 0$$

When $\alpha = 0$ there is a polynomial solution to this system: $V = aY^2 + bY + c$, $v = KY + k$ where a, b, c, K, k are functions of time with $a(T) = b(T) = c(T) = \dot{K}(T) = K(T) = \dot{k}(T) = k(T) = 0$. By identification for all Y it is found that ($X_d = 1 \ \Rightarrow\ Y_d = 0$):

$$\begin{aligned} &\dot{a} + 2aK + \beta\dot{K}^2 + 1 = 0, \quad \dot{b} + Kb + 2ak + 2\beta\dot{K}\dot{k} = 0, \\ &\dot{c} + kb + \sigma^2 a + \beta\dot{k}^2 = 0, \quad \beta\ddot{K} + a = 0, \quad 2\beta\ddot{k} + b = 0, \end{aligned} \tag{56}$$

One must also check that $u_t = r - \frac{\sigma^2}{2} - \kappa X_t - K(t)\log X_t - k(t)$ is in $\mathcal{U}$.
A solution procedure is to use the next to last equation in the first one and the last one into the second one:

$$\begin{aligned} &-\dddot{K} - 2K\ddot{K} + \dot{K}^2 + \tfrac{1}{\beta} = 0, \quad \ddot{K}(T) = \dot{K}(T) = K(T) = 0 \\ &-\dddot{k} - K\ddot{k} + \dot{K}\dot{k} - \ddot{K}k = 0, \quad \ddot{k}(T) = \dot{k}(T) = k(T) = 0 \ \Rightarrow k(t) = 0\ \forall t. \end{aligned} \tag{57}$$

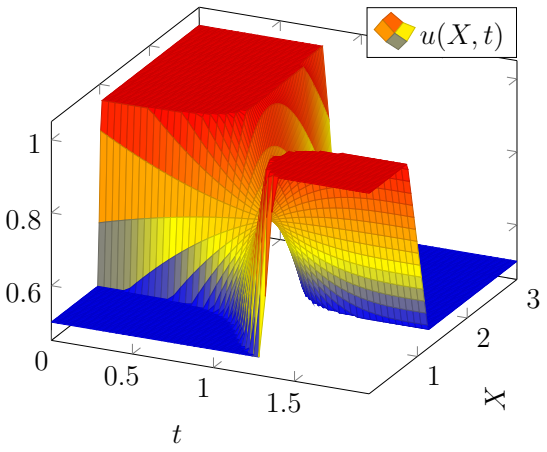

Figure 19: Solution of the dynamic programming equation: side view.

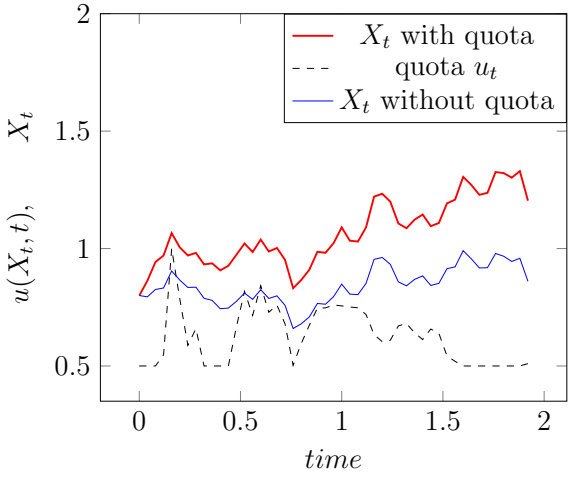

Figure 20: Optimal biomass and quota function computed by Optimal Control and compared with the solution with $u = u_M$ when $X_0 = 0.8$.