



HAL
open science

Dynamic Programming versus supervised learning

Gilles Pagès, Olivier Pironneau

► **To cite this version:**

Gilles Pagès, Olivier Pironneau. Dynamic Programming versus supervised learning. Numerical Control: Part A, 23, Elsevier, pp.467-497, 2022, Handbook of Numerical Analysis, 10.1016/bs.hna.2021.12.014 . hal-03891198

HAL Id: hal-03891198

<https://cnrs.hal.science/hal-03891198>

Submitted on 22 Jul 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

Chapter 1

Dynamic Programming versus Supervised Learning

Optimal fishing quotas

Gilles Pagès^a and Olivier Pironneau^b

^a *gilles.pages[at]sorbonne-universite.fr, LPMA, Sorbonne Université, Paris, France,*

^b *olivier.pironneau[at]sorbonne-universite.fr, LJLL, Sorbonne Université, Paris, France.*

Chapter Points

- This chapter deals with the design of optimal fishing quota within a model used in Senegal for the long term evolution of the fish biomass.
- The numerical tests validate supervised learning for stochastic control in this case. However the control is not Markovian, but for fishing quotas this does not seem to be a problem .
- For comparison the same problem is solved with Dynamic Programming. Three implementations are tested using either a probabilistic method or the Hamilton-Jacobi partial differential equations.

1.1 INTRODUCTION

Stochastic control of differential systems with random data or modeling errors, is usually solved by Dynamic Programming, a technique introduced in the 1950 by Richard Bellman (3). 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 (14), (12), 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 several years ago (see (2), (11), (4), (1) and others): Supervised Learning with Neural Networks: like off-line/on-line reduction methods, one is ready to pay a high CPU price in a "training" 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 chapter the three classes of methods will be tested on an easy yet practical problem studied in (6): the control of a fishing site in Senegal where the fish biomass is threatened by intensive fishing. In (1) it was shown that parameter identification from a few measurements near time zero could be obtained

accurately and fast by supervised learning. Here the parameters are known but with some uncertainty and the objective is to apply optimal quotas to stabilise 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 .

Quotas are meant to keep the biomass at a constant desired level X_d . Hence the total number of fish caught must be lower than the quota Q_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. Nevertheless each boat is told to keep their catch below Q_t/E_t .

The control problem is solved by minimizing a criteria, the integral over time of $\|X_t - X_d\|^2$. But quotas, imposed by the regulator, are unpopular. To avoid too low (constraining) quotas, we add to the criteria the integral over time of $-\alpha Q_t$ where α is constant and positive; consequently, a low quota will not be optimal because it gives a larger value to the criteria than a big quota.

Along the way we found that the solution of the problem is very often "bang-bang", i.e; switching from the structural limit to the quota often. For the regulator, rapidly changing quotas may be also dangerously unpopular; hence in an attempt to avoid such frequent drastic changes we add to the criteria the integral over time of $\beta\|\dot{Q}_t\|^2$ where β is a positive constant.

More details on the modelling of the problem can be found in Appendix A.

The plan of the paper is as follows:

In section 1, the 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 formulations:

1. Deterministic control where the state is the solution of an *ODE*.
2. Non-dynamic control, i.e. an *SDE* for the state equation but a deterministic control function of time only.
3. A fully stochastic state and control, usual to 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. We have used the standard multilayered Recurrent Neural Networks of Keras (7) (RNN, see figure 1.3); LSTM is also available in Keras, but as there are no repeating patterns in the solution of a *SDE* with brownian noise, it is not clear that a Long-Short-Term-Memory feedback architecture would improve the (already good) performance of RNN.

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. It will be seen that these methods do not handle very well the penalty term $\beta\|\dot{Q}_t\|^2$.

1.2 A MODEL PROBLEM

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

$$\min_{u \in \mathcal{U}} J(u) := \int_0^T \mathbb{E} [(X_t - X_d)^2 - \alpha u(X_t, t) + \beta |\partial_t u(X_t, t)|^2] dt, \quad (1.1)$$

$$\text{with } dX_t = X_t(r - \kappa X_t - u_t)dt + \sigma X_t dW_t, \quad X_0 = X^0 \geq 0, \quad (1.2)$$

where W_t is a standard Wiener process on a probability space $(\Omega, \mathcal{A}, \mathbb{P})$, $X_d, r, \kappa, \sigma, u_m, u_M$ are real constants which depend on the considered fishing site and

$$\mathcal{U} = \{u \in H^1(]0, T[, W^{1,1}) : u_m \leq u(t, X_t) \leq u_M \text{ a.s.}\}.$$

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" $E(t)$, modeled by

$$dE_t = (a - u_t X_t - cE_t) dt, \quad E_0 = E^0 \text{ given,}$$

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 aims at keeping a biomass at a desired level X_d at all times, but, quotas being unpopular, there is a term to prevent low quotas, $-\alpha u_t$ and another to prevent quotas from changing too quickly: $\beta |\partial_t u|^2$.

It is important to notice that one may also consider the case where u is a deterministic function of t only:

$$\min_{u \in \mathcal{U}_d} J(u) := \int_0^T \mathbb{E} [(X_t - X_d)^2] - \alpha u(t) + \beta |\partial_t u(t)|^2 dt \text{ with (1.2),} \quad (1.3)$$

$$\text{with } \mathcal{U}_d = \{u \in H^1(]0, T[) : u_m \leq u(t) \leq u_M\}.$$

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 $\sigma = 0$.

1.2.1 Existence of solution: the deterministic case

In the deterministic case the problem reads

$$\min_{u \in \mathcal{U}_d} \left\{ J(u) = \int_0^T |X - X_d|^2 - \alpha u + \beta |\dot{u}|^2 : \frac{\dot{X}}{X} = r - \kappa X - u, X(0) = X^0 \right\}. \quad (1.4)$$

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

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

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

$$\begin{aligned} \dot{X}_i &= rX_i - \kappa X_i^2 - u_i X_i, \quad i = 1, 2, \Rightarrow \\ \dot{X}_1 - \dot{X}_2 &= (X_1 - X_2)(r - \kappa(X_1 + X_2) - (u_1 + u_2)) - (u_1 - u_2)(X_1 + X_2) \Rightarrow \\ |X_1(t) - X_2(t)| &\leq \alpha \int_0^t |X_1(s) - X_2(s)| ds + 2X_M \int_0^t |u_1(s) - u_2(s)| ds \end{aligned}$$

so that by a Grönwall argument, one derives that, for every $t \in [0, T]$

$$|X_1(t) - X_2(t)| \leq 2X_M e^{\alpha T} \|u_1 - u_2\|_{L^2([0, T])}.$$

This also shows that $u \mapsto \int_0^T |X(t) - X_d|^2 dt$ is continuous on $L^2([0, T])$.

Proposition 1. *Problem (1.4) has a solution.*

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

$$J(u^*) \leq \liminf J(u^n) = \inf_{u \in \mathcal{U}_d} J(u) \Rightarrow u^* \text{ is a solution.}$$

1.2.2 Existence of solution: the stochastic non-dynamic case

Proposition 2. *The solution of the SDE (1.2) exists and is positive.*

Proof. $X \mapsto (r - \kappa X - u_t)X$ 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 $\mathcal{F}_t^W = \sigma(\mathcal{N}_s, W_s, s \leq t)$ where \mathcal{N}_s denotes the \mathbb{P} -negligible sets of \mathcal{A} . On $[0, \tau[$ we may work with $Y_t = \log X_t$, solution (by Itô calculus) of

$$\begin{aligned} dY_t &= \left(r - \frac{\sigma^2}{2} - \kappa X_t - u_t \right) dt + \sigma dW_t, \quad \Rightarrow \quad \forall t \in (0, \tau] : \\ X_t &= X^0 \exp \left(\left(r - \frac{\sigma^2}{2} \right) t - \int_0^t (\kappa X_s + u_s) ds + \sigma W_t \right) \leq X^0 e^{\mathcal{S}_t(u_m)}, \end{aligned} \quad (1.5)$$

where $S_t(v) := (r - \frac{\sigma^2}{2})t - vt + \sigma W_t$. Hence $\int_0^\tau X_s ds = +\infty$ is impossible unless $\tau = +\infty$, \mathbb{P} -a.s. Therefore, $X_t \in \mathbb{R}^+$ for every $t \geq 0$ and

$$X^0 e^{S_t(u_M)} \exp\left(-\kappa \int_0^t e^{S_s(u_M)} ds\right) \leq X_t \leq X^0 e^{S_t(u_m)}. \quad (1.6)$$

Theorem 1. *Problem (1.1)(1.2) has a solution.*

Proof. Consider two trajectories driven by the same W_t but with different u , denote δX_t the difference and \bar{X}_t their average; we have

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

By Itô calculus:

$$\begin{aligned} (\delta X_t)^2 &= 2 \int_0^t ((\delta X_s)^2 (r + \frac{\sigma^2}{2} - 2\kappa \bar{X}_s - \bar{u}) - \bar{X}_s \delta u(s) \delta X_s) ds + 2 \int_0^t \sigma (\delta X_s)^2 dW_s \\ \text{so that } \mathbb{E}[(\delta X_t)^2] &\leq 2(r + \frac{\sigma^2}{2} - u_m) \int_0^t \mathbb{E}[(\delta X_s)^2] ds - \int_0^t \delta u(s) \mathbb{E}[\bar{X}_s \delta X_s] ds \\ &\leq 2(r + \frac{\sigma^2}{2} - u_m) \int_0^t \mathbb{E}[(\delta X_s)^2] + \int_0^t \delta u(s) \mathbb{E}[(\bar{X}_s)^2] ds. \end{aligned}$$

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

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

and by Grönwall's lemma again, $\mathbb{E}[(\delta X_t)^2] \leq 4e^{2r + \sigma^2 - 2u_m)t} e^{(2r + \sigma^2)T} \mathbb{E}[X_0^2] \int_0^t |\delta u(s)|$. Hence, there exists $C > 0$, $\sup_{t \in [0, t]} \mathbb{E}[(\delta X_t)^2] \leq C \mathbb{E}[X_0^2] \|\delta u\|_{L^1([0, T])}$. Continuity of $u \mapsto \int_0^T \mathbb{E}[(X_t - X_d)^2] dt$ follows and existence is then shown as in (1.2.1).

1.2.3 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 1.7 that there is a solution when the \mathbb{E} in the criteria and κ are multiplied by a cutoff for large X and, with $\mathcal{Q}^+ := \mathbb{R}^+ \times]0, T[$, and a modified \mathcal{U} :

$$\mathcal{U} = \{u \in L^2(W^{1,1}(\mathbb{R}^+)) : u \in [u_m, u_M], \partial_t u \in L^2(\mathcal{Q}^+), \lim_{X \rightarrow +\infty} |\log X| u(X, t) < +\infty\}.$$

1.2.4 Discretization

Let $h = \frac{T}{M}$ and $t^m = mh$. For any f , f^m denotes an approximation of $f(mh)$. Define the Euler – or Euler-Maruyama – scheme of the diffusion by

$$X^{m+1} = X^m (1 + h(r - \kappa X^m - u^m) + \sigma \delta W^m) \quad (1.7)$$

where $\delta W^m = W_{(m+1)h} - W_{mh}$. Note that positivity of X may not be preserved by this scheme, but we may choose to discretize (1.5). Alternatively, to avoid the computing cost of exponentials, we may use the Milstein scheme:

$$\tilde{X}^{m+1} = \tilde{X}^m \left(1 + h(r - \kappa \tilde{X}^m - u^m - \frac{\sigma^2}{2}) + \sigma \delta W^m + \frac{\sigma^2}{2} (\delta W^m)^2 \right). \quad (1.8)$$

Proposition 3. *Let \tilde{X} denote the solution of the continuous Milstein scheme given for every $t \in [mh, (m+1)h)$ by*

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

Assume $\tilde{X}_0 = X^0 > 0$. Then, on event $A_h = \left\{ \tilde{X}^m < \frac{1}{2\kappa h} + \frac{r - \sigma^2/2 - u_M}{\kappa} \right\}_{m=0}^{M-1}$, the continuous Milstein $(\tilde{X}_t)_{t \in [0, T]}$ almost surely has positive paths.

Proof. We proceed by induction. Starting from (1.8), we want the discriminant of the second degree polynomial equation in $W_t - W^m$ to be negative, namely after obvious simplification by $\sigma > 0$

$$1 - 2 \left(1 + h(r - \kappa \tilde{X}^m - u^m - \frac{\sigma^2}{2}) \right) < 0 \iff \tilde{X}^m < \frac{1}{2\kappa h} + \frac{r - \sigma^2/2 - u_M}{\kappa}.$$

Remark 1. *It is clear that $\mathbb{P}(A_h) \uparrow 1$ as $h \rightarrow 0$. For more details on the positivity preserving property of the Milstein scheme, see (15, Chapter 7.5).*

1.3 BRUTE FORCE SOLUTION OF THE NON-DYNAMIC CONTROL PROBLEM BY MONTE-CARLO

We wish to solve (a discretized version of) Problem (1.3) by a Monte Carlo simulation. Recall that $\delta W^m = \sqrt{h} \mathcal{N}_{0,1}^m$ where $(\mathcal{N}_{0,1}^m)_{m=1, \dots, M}$ denotes the numerical realization of an i.i.d. sequence of $\mathcal{N}(0, 1)$ -distributed random variables. For each of the above three schemes there is a function ζ_h such that

$$X^{m+1} = \zeta_h(X^m, u^m, \mathcal{N}_{0,1}^{m+1}), \quad 0 \leq m \leq M-1, \quad X^0 \text{ given.} \quad (1.9)$$

Let $(\mathcal{N}_{0,1,k}^m)_{k \in [1, K]}^{m \in [0, M]}$, be $K \times M$ independent realizations of $\mathcal{N}_{0,1}$ and, for $\{u^m \in [u_m, u_M]\}_{m=1}^M$ given, independent of k , let $\{X_k^m\}_{k=1}^K$ be the K (independent) results of (1.9). By the law of large numbers:

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

Then (1.3) can be approximated by

$$\min_{\{u^m \in [u_m, u_M]\}_{m=0}^{M-1}} J_K := h \sum_{m=0}^{M-2} \left[\mathbb{E}_K [(X^{m+1} - X_d)^2] - \alpha u^m + \frac{\beta}{h^2} |u^{m+1} - u^m|^2 \right]. \quad (1.10)$$

1.3.1 Solution with a gradient method

Gradient methods update $\{u^m\}_{m=1}^{M-1}$ 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 (1.7); let us differentiate this equation and multiply it by an adjoint variable P_k^m and sum:

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

As $\delta X^0 = 0$, if we set $P^{M-1} = P^{M-2} = 0$, $v_k^m = 1 + h(r - 2\kappa X_k^m - u^m) + \sigma \delta W_k^m$,

$$- \sum_0^{M-2} h P_k^m X_k^m \delta u^m = \sum_1^{M-2} (P_k^{m-1} \delta X_k^m - P_k^m \delta X_k^m v_k^m) = \sum_1^{M-2} \delta X_k^m (P_k^{m-1} - P_k^m v_k^m).$$

Define P_k^m by $P_k^{m-1} = P_k^m v_k^m - 2(X_k^m - X_d)$, $m = M-2, M-3, \dots$. Then

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

So the gradient is $G^m = \frac{1}{K} \sum_{k=1}^K P_k^m X_k^m - \alpha - 2 \frac{\beta}{h^2} (u^{m+1} - 2u^m + u^{m-1})$. Note

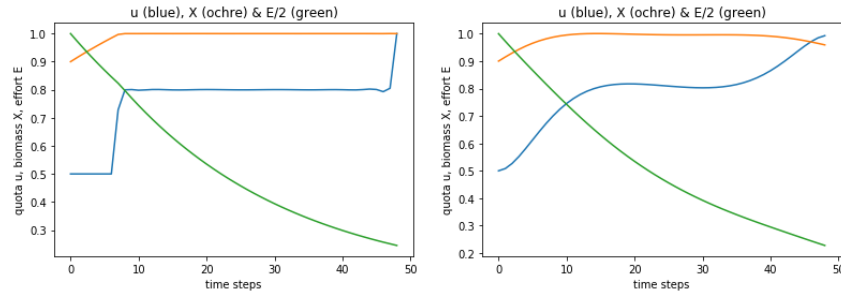


FIGURE 1.1 Solution of (1.1) discretized by (1.10), versus m , in the deterministic case ($\sigma = 0$, $K = 1$) with $\beta = 0$ (left), and $\beta = 0.001$ (right). The blue curve is the control u ; the biomass X_t is in ochre and the fishing effort in green.

that the successive $(u^m)_m = 0, \dots, M-1$ obtained by this gradient descent are not adapted and depend on all the $M \times K$ simulated $\mathcal{N}_{0,1,k}^m$.

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 and an automatic adjustment of the step size μ . We took 50 time steps and chose to solve a deterministic case with

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

The computer program is in file `fishingQuotaBruteforce.py`. The results with $\alpha = \beta = 0$ are shown on the left in figure 1.1 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 $X^{m+1} = X^m = X_d, P^m = 0$. On the right in figure 1.1 the results are shown for $\alpha = 0.01, \beta = 0.001$. On figure 1.2 a stochastic case of (1.10) is solved with the same parameters except $\sigma = 0.3$. The results are shown for $K = 1, 10, 100, 10000$ Monte-Carlo samples.

1.4 SOLUTION OF THE NON-DYNAMIC CONTROL PROBLEM BY SUPERVISED LEARNING

We still aim at to solving (1.3) using Monte Carlo simulations (written here in continuous time for genericity)

$$\min_{u \in \mathcal{U}} J(u) := \int_0^T \left[\frac{1}{K} \sum_{k=1}^K (X_k(t) - X_d)^2 - \alpha u(t) + \beta (\partial_t u(t))^2 \right] dt \quad (1.11)$$

$$\text{with } dX_k = X_k(t)(r - \kappa X_k(t) - u(t))dt + \sigma X_k(t)dW_k(t), \quad X(0) = X^0. \quad (1.12)$$

where W^k are K independent standard Brownian motions X_k is replaced by a numerical scheme. In practice To reduce CPU cost, we shall train a neural network with input parameters $[X^0, r, \kappa]$ and output an optimal $\{u^m\}_{m=1}^{M-1}$ 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. We have used a small $K = 100$ 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 $[X_m^0, X_M^0] \times [r_m, r_M] \times [\kappa_m, \kappa_M]$.
- Generate $K \cdot M$ normal gaussian random values for dW_t .
- Compute a discrete optimal solution $\{u^m\}_{m=1}^{M-1}$ of (1.11) with $K=100$ by the L-BFGS-B Python module.
- Then train a neural network $[X^0, r, \kappa] \rightarrow \{u^m\}_{m=1}^{M-1}$ with these 10000 synthetic solutions.

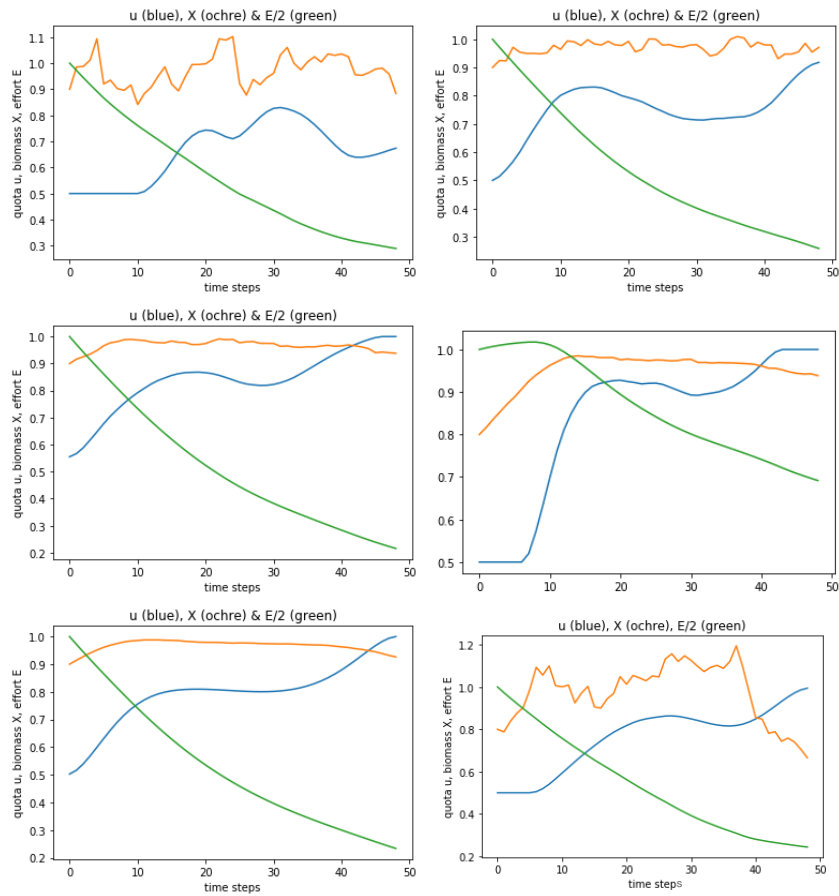


FIGURE 1.2 Solution of (1.10) in the stochastic case with $\sigma = 0.3$: convergence of the Monte-Carlo approximation. Left to right, up first: $K=1$, $K=10$, $K=100$, $K=1000$, $K=10000$. Already the results are fairly converged at $K=100$. In the last figure the optimal control is computed with $K=100$ and used to solve (1.2) with a new random noise, thereby showing that the solution is still acceptable. Note that the brute force solution for $K \gg 1$ is quite close to the noiseless solution of figure 1.1-b.

- Finally, check the precision by using the network's answer $\{u^m\}_{m=1}^{M-1}$ 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 (8). We have used Keras (see (7)) to generate a 4-layer Neural Network (see figure 1.3). 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.

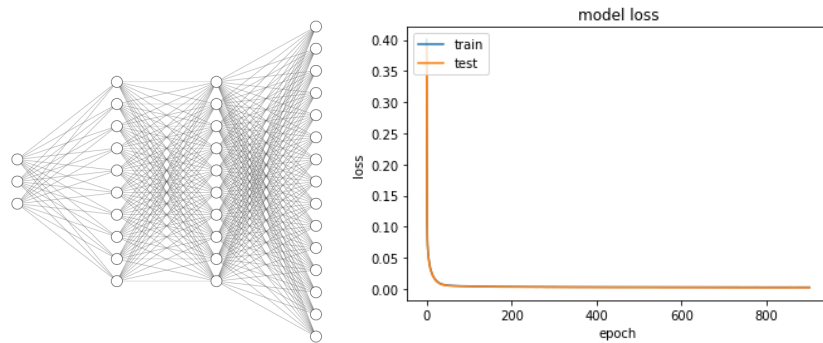


FIGURE 1.3 The Neural Network has 3 inputs, two hidden layers each with 50 neurons (10 shown here) and separated by batch-normalisation and an output layer with M output (M=15 shown here). Right: typical convergence curve of the loss function during the training phase.

1.4.1 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 10 000 samples in batches of 32 and a maximum of 2 000 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)
```

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

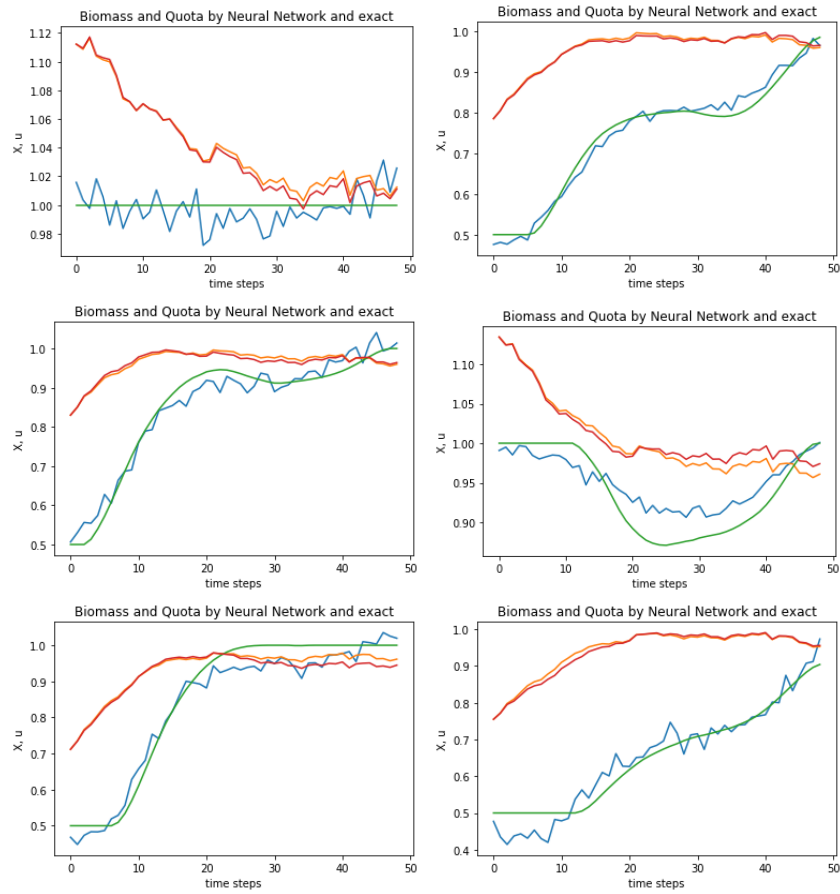


FIGURE 1.4 On each of the 6 plots X and u are shown: True discrete solution in green for X and red for u . NN solution in blue for X_{NN} and ochre for u_{NN} . The plots correspond to the following values of the parameters: $r = 2.02285$, $\kappa = 1.0203$, $X_0 = 1.1122$, $r = 2.0688$, $\kappa = 1.2641$, $X_0 = 0.7852$, $r = 1.9151$, $\kappa = 1.0141$, $X_0 = 0.8299$, $r = 2.0738$, $\kappa = 1.1876$, $X_0 = 1.1344$, $r = 2.0463$, $\kappa = 1.1224$, $X_0 = 0.7550$.

Average relative error $|1-u_{NN}/u|$ for the 49 time steps =
 [0.031 0.03954 0.04612 0.03440 0.03665 0.0432 0.03549 0.04364
 0.04247 0.03737 0.03448 0.03950 0.04441 0.04149 0.03366 0.04054
 0.03978 0.04160 0.0404 0.03319 0.03800 0.03229 0.04076 0.03475
 0.03333 0.03761 0.04427 0.03837 0.03909 0.03384 0.03707 0.03838
 0.03682 0.03278 0.03412 0.02993 0.03138 0.02646 0.02687 0.02550
 0.02197 0.01949 0.02680 0.02290 0.01801 0.01872 0.01909 0.014978 0.01907]

Notice that the error is about the same at all time steps.

In figure 1.4 the performance of the trained Neural Network 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 then, X_{NN} computed by (1.7) with u_{NN} is close to X computed with u “exact”. This reveals that X is not that sensitive to u : that the problem is somewhat stiff.

1.5 BELLMAN'S STOCHASTIC DYNAMIC PROGRAMMING FOR THE DYNAMIC PROBLEM

Consider the *value function*

$$V(t, X) = \min_{u \in \mathcal{U}} \left\{ \int_t^T \mathbb{E} [(X_\tau - X_d)^2 - \alpha u + \beta |\partial_t u|^2] d\tau : \right. \\ \left. dX_\tau = X_\tau d\tau (r - \kappa X_\tau - u_\tau) + \sigma X_\tau dW_\tau, X(t) = X \right\}. \quad (1.13)$$

Let $[0, T] = \cup_1^M [(m-1)h, mh]$. Let $\zeta_h(X, u^m, z)$ denote one iterate of a numerical scheme (1.7)(1.8) for the *SDE* starting at X , for instance with (1.7)

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

Let $v^M(X) = 0$ and let, for $m \in \{0, \dots, M-1\}$, $v^m(X) =$

$$\min_{u \in \mathcal{U}} \int_{mh}^{(m+1)h} \mathbb{E} [|\zeta_h(X, u, z) - X_d|^2 - \alpha u + \beta |\partial_t u|^2] d\tau + \mathbb{E} [v^{m+1}(\zeta_h(X, u, z))] \\ \approx \min_{u \in \mathcal{U}} \left\{ h \mathbb{E} [|\zeta_h(X, u, z) - X_d|^2] - \alpha hu + \frac{\beta}{h} |u^{m+1}(X) - u|^2 + \mathbb{E} [v^{m+1}(\zeta_h(X, u, z))] \right\}. \quad (1.14)$$

Evidently $\mathbb{E} [|\zeta_h(X, u, z) - X_d|^2] = X - X_d + hX(r - \kappa X - u)^2 + hX^2\sigma^2$. To compute $\mathbb{E} [v^{m+1}(\zeta_h(X, u, z))]$, we use a quadrature formula with Q points $\{z_q\}_{q=1}^Q$ and weights $\{w_q\}_{q=1}^Q$ based on optimal quantization of the normal distribution $\mathcal{N}(0, 1)$ (see (16) or (15, Chapter 5) and the website quantize.maths-fi.com)

for download of weighted grids) so that

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

Finally at every time step and every $X_j = jL/J$, $j = 0, \dots, J$, with $L \gg 1$, 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 1.5, 1.6 have been obtained. Once $\{u^m(X)\}_0^{M-1}$

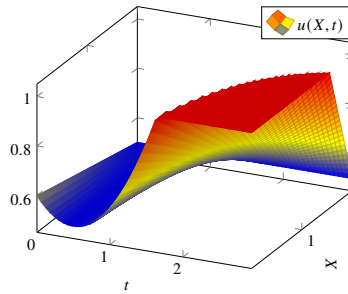


FIGURE 1.5 Solution of the Dynamic Programming equation: $u(x, t)$.

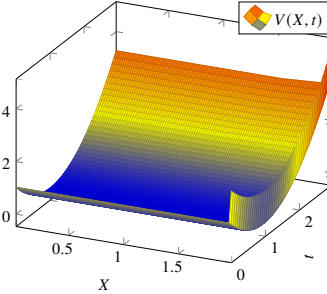


FIGURE 1.6 Solution of the Dynamic Programming equation: $V(x, t)$.

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 1.7, 1.8. 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 A which deals with a very similar problem for which there is a polynomial in $\log X$ solution when $\alpha = 0$.

1.6 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 (1.13).

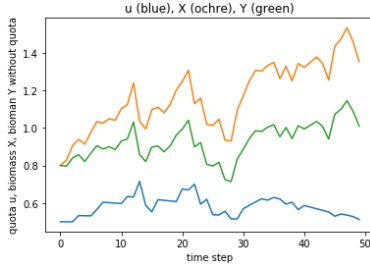


FIGURE 1.7 Simulation of the fishing model with a quota function computed by Dynamic Programming and $X_0 = 0.8$.

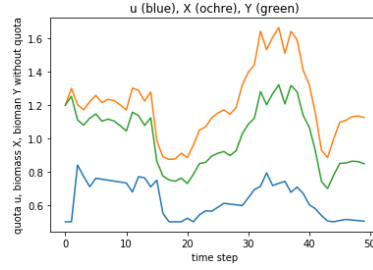


FIGURE 1.8 Simulation of the fishing model with a quota function computed by Dynamic Programming and $X_0 = 1.2$.

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

$$0 = \inf_{u \in \mathcal{U}} \left\{ (X - X_d)^2 - \alpha u + \beta |\partial_t u|^2 + \partial_t V + (r - \kappa X - u) X \partial_X V + \frac{\sigma^2 X^2}{2} \partial_{XX} V \right\}. \quad (1.15)$$

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 (10). We return to (1.14) above. 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

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

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

$$v^m(X) \approx \min_{u \in \mathcal{U}} \left\{ \left(|X - X_d|^2 - \alpha u + \frac{\beta}{h^2} |u^{m+1}(X) - u|^2 \right) h + \mathbb{E} [v^{m+1}(\zeta_h(X, u, z))] \right\}. \quad (1.16)$$

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

$$\begin{aligned} \mathbb{E} [v^{m+1}(\zeta_h(X, u, z))] &= v^{m+1}(X) + \mathbb{E} \left[\sum_0^{q_M} v^{(q)m+1} X^q h^{\frac{q}{2}} (\sigma z + \lambda \sqrt{h})^q \right] \approx v^{m+1}(X) \\ &+ X v'^{m+1} \lambda h + \frac{X^2}{2} v''^{m+1} (\lambda^2 h + \sigma^2) h + \frac{X^3}{6} \lambda \sigma^2 v^{(3)m+1} h^2 + \dots \end{aligned}$$

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

Let us use it in (1.16). Then u must solve

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

but comply also with the box constraints:

$$u^m = \min \left\{ u_M, \max \left\{ u_m, u^{m+1} + \frac{h^2}{2\beta} (\alpha + X v'^{m+1}) \right\} \right\} \text{ and}$$

$$v^m \approx v^{m+1} + h \left(X v'^{m+1} (r - \kappa X - u^m) + \frac{X^2 \sigma^2}{2} v''^{m+1} + |X - X_d|^2 - \alpha u^m + \frac{\beta}{h^2} |u^{m+1} - u^m|^2 \right).$$

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 mesh size.

1.6.1 Numerical Results

We have implemented the method using FreeFEM++ (9) with a slight change : 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 numerical stability.

Numerical results shown on figure 1.9, 1.10 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.

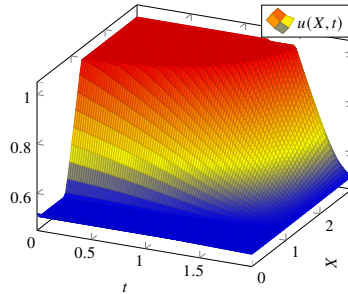


FIGURE 1.9 Solution of the Dynamic Programming equation: side view.

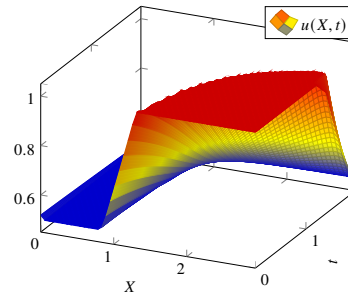


FIGURE 1.10 Solution of the Dynamic Programming equation: front view.

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 1.11,1.12.

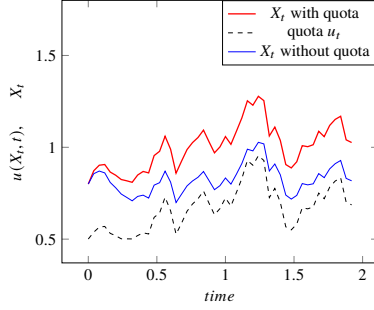


FIGURE 1.11 Simulation of the fishing model with a quota function computed by Dynamic Programming and $X_0 = 0.8$.

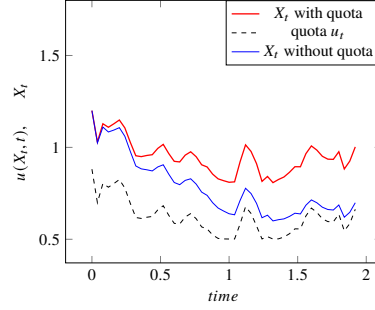


FIGURE 1.12 Simulation of the fishing model with a quota function computed by Dynamic Programming and $X_0 = 1.2$.

1.6.2 Analysis when $\beta = 0$

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

Proof. With $\mathcal{U} = \{u = (u_m)_n, u_m \in [u_m, u_M]\}$ 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

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

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

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

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

Remark 2. *Note that this HJB dynamic solution is different from the non-dynamic solution of figure 1.1.*

1.6.3 Discretization

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

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

The stationary point is at

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

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 1.13, 1.14. The control is indeed bang-bang.

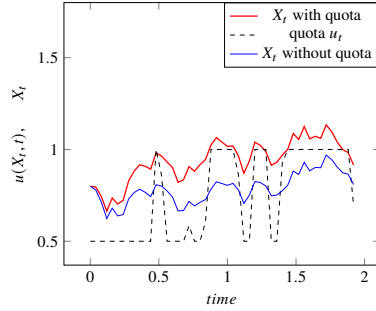


FIGURE 1.13 Simulation of the fishing model with a quota function computed by HJB when $\beta = 0$ and $X_0 = 0.8$.

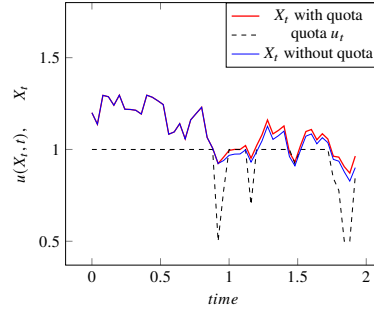


FIGURE 1.14 Simulation of the fishing model with a quota function computed by HJB when $\beta = 0$ and $X_0 = 1.2$.

1.7 SOLUTION WITH THE KOLMOGOROV EQUATION

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

$$\partial_t \rho + \partial_X [(r - \kappa X - u)X\rho] - \partial_{XX} [\frac{X^2 \sigma^2}{2} \rho] = 0, \quad \rho(X, 0) = \rho^0(X), \quad \forall X \in \mathbb{R}^+. \quad (1.17)$$

The optimal control problem (1.1) is: with (1.17)

$$\min_{u \in \mathcal{U}} J(u) := \int_{Q^+} [(X - X_d)^2 - \alpha u(X, t) + \beta |\partial_t u(X, t)|^2] \rho(X, t) dX dt. \quad (1.18)$$

The conditions for having equivalence between the stochastic control written as (1.1) and (1.18) are detailed in (13). 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 $dY_t = v dt + \sigma dW_t$ and then study the existence of a PDE for Y_t given by

$$\partial_t \rho + \partial_Y (v\rho) - \partial_{YY} [\frac{\sigma^2}{2} \rho] = 0, \quad \rho(Y, 0) = \rho^0(Y), \quad \forall Y \in \mathbb{R}, \quad t \in]0, T[. \quad (1.19)$$

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

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

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

Proposition 6. *If $v \in W_{loc}^{1,1}(\mathbb{R})$, $v/(1+|y|) \in L^1(\mathbb{R}) \cap L^\infty(\mathbb{R})$, $\partial_y v \in L^\infty(\mathbb{R})$ then Y_t has a PDF, $\rho \in L^\infty(L^2(\mathbb{R}) \cap L^\infty(\mathbb{R})) \cap L^2(H^1(\mathbb{R}))$, given by (1.19).*

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 $X \rightarrow +\infty$. 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 $\int_{\mathbb{R}} \rho^0(u) du = 1$, let $Q = \mathbb{R} \times]0, T[$ and $Q_R =]-R, R[\times]0, T[$ and,*

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

$$\min_{v \in \mathcal{V}} J(v) := \int_{Q_R} [(e^Y - X_d)^2 + \alpha v(Y, t) + \beta |\partial_t v(Y, t)|^2] \rho(Y, t) dY dt$$

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

Problem (1.20) 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 $\tilde{\rho}(Y, t) = (4\pi)^{-\frac{1}{2}} \int_{\mathbb{R}} e^{-\frac{X^2}{4t}} \rho^0(X - Y) dX$. 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

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

Consequently, and because $\|v\|_\infty \leq v_M$, ρ is bounded in $L^2(H^1(\mathbb{R})) \cap L^\infty(L^2(\mathbb{R})) \cap L^2(Q)$ and $\partial_t \rho$ is bounded in $L^2(H^{-1}(\mathbb{R}))$. Consequently ρ is compact in $L^2(Q)$ when v varies in \mathcal{V} . From the maximum principle we also know that $0 \leq \rho_n \leq 1$ in Q and that $\int_{\mathbb{R}} \rho(Y, t) dY = 1$ a.e.

Now let us take a minimizing sequence $\{v_n\}_{n \in \mathbb{N}}$ for (1.20). Then ρ_n is bounded in $L^2(H^2(\mathbb{R})) \cap L^\infty(L^2(\mathbb{R})) \cap W^{1,2}(H^{-1}(\mathbb{R}))$ and compact in $L^2(Q_R)$. It follows from above that there is a v such that, for a subsequence, $v_n \rightarrow v$ in L^∞ -weak-star and weakly in $L^2(Q)$ and a ρ such that $\rho_n \rightarrow \rho$ strongly in $L^2(Q)$. Hence $\int_{\mathbb{R}} v_n \rho_n \partial_Y \hat{\rho} \rightarrow \int_{\mathbb{R}} v \rho \partial_Y \hat{\rho}$ for any $\hat{\rho} \in L^2(H^1(\mathbb{R}))$ and consequently ρ solves the PDE with v .

As $\rho_n \in [0, 1]$ and by Fatou's lemma, $\int_{Q_R} (e^Y - X_d)^2 \geq \liminf \int_{Q_R} (e^Y - X_d)^2 \rho_n \geq \int_Q (e^Y - X_d)^2 \rho$. By the weak (resp. strong) convergence of v_n (resp. ρ_n), $\int_{Q_R} \alpha v_n \rho_n \rightarrow \int_{Q_R} \alpha v \rho$.

To analyze the last term in the criteria we write it as

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

By the lower semi-continuity of semi-norms, $\lim_{n \rightarrow \infty} \int_{Q_R} |\partial_t v_n|^2 \rho \geq \int_{Q_R} |\partial_t v|^2 \rho$. Finally the last term above is bounded by $\beta \|(\partial_t v_n)^2\|_{L^2(Q_R)} \|\rho_n - \rho\|_{L^2(Q_R)} \leq \beta K \|\rho_n - \rho\|_{L^2(Q_R)} \rightarrow 0$. Consequently, the criteria in (1.20) is l.s.c.

Remark 3. If $\int_{\mathbb{R}^+} X^2 \rho^0(X) dX < \infty$, it may be possible to let $R \rightarrow \infty$.

1.7.1 Computation of gradients

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

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

Calculus of variations (see (??) for details) leads to $\delta \rho(0) = 0$ and

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

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

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

Add the two equations above 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

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

Finally, from (1.18),

$$\begin{aligned} \delta J &= \langle \text{grad}_u J, \delta u \rangle \\ &= \int_{\Omega \times]0, T[} \left[\left((X - X_d)^2 - \alpha u + \beta |\partial_t u|^2 \right) \delta \rho - \alpha \rho \delta u + 2 \rho \partial_t u \partial_t \delta u \right] \\ &= - \int_{Q^+} \rho X \partial_X p' \delta u + \int_{\Omega \times]0, T[} \rho [-\alpha \delta u + 2 \partial_t u \partial_t \delta u]. \end{aligned}$$

1.7.2 Results

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

$$v = u^k - \mu \text{grad}_u J^k, \quad u^{k+1} = \max\{u_m, \min\{u_M, v\}\}. \quad (1.20)$$

Note that we have constrained u to satisfy $\partial_t u(0) = \partial_t u(T) = 0$ so as to integrate by parts the term $\partial_t u \partial_t \delta u$. This is only a slight loss of generality. 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 1.15 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 1.16 and 1.17 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 was not playing its regularizing role.

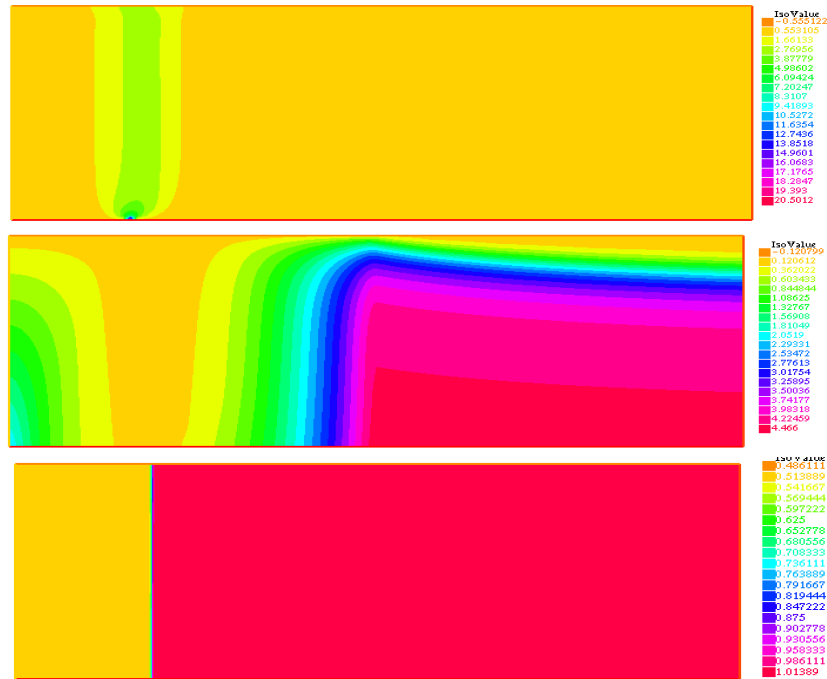


FIGURE 1.15 From left to right: optimal ρ, p', u . Horizontal axis is $X \in]0, X_M[$, vertical is $t \in [0, T]$ with $X_M = 5, T = 2, X_d = 1, u_m = 0.5, u_M = 1$. Note that $(X, t) \mapsto u(X, t)$ is either 0.5 or 1, implying that all controls $u(X_t, t)$ will be bang-bang. There is a vertical scale factor of 0.5 applied.

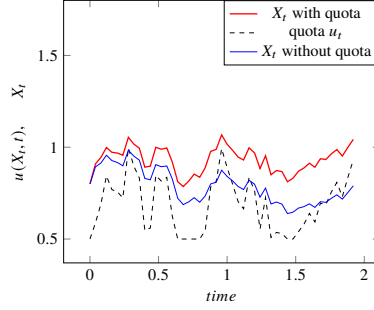


FIGURE 1.16 Optimal biomass and quota function computed by Optimal Control and compared with the solution with $u = u_M$ when $X_0 = 0.8$.

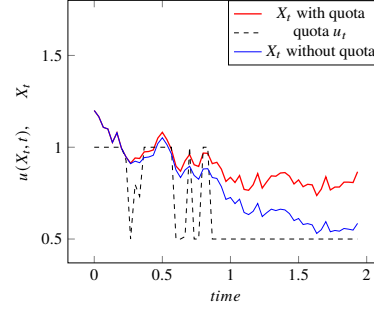


FIGURE 1.17 Optimal biomass and quota function computed by Optimal Control and compared with the solution with $u = u_M$ when $X_0 = 1.2$.

1.8 SOLUTION BY ITÔ CALCULUS

In computational finance Itô calculus is preferred over the Kolmogorov setting, primarily because the initial condition for the PDF are singular.

Consider again Problem (1.1). Itô calculus applied to

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

(1) yields

$$\partial_t j + (r - \kappa X - u)X \partial_X j + \frac{X^2 \sigma^2}{2} \partial_{XX} j = -[(X - X_d)^2 + \beta(\partial_t u)^2 - \alpha u]. \quad (1.21)$$

Let us compute j as the solution at $\tau = 0$ and $X = X^0$ of the *PDE*, backward in time and initiated by $j(X, T) = 0, \forall X \in \mathbb{R}^+, \tau \in]0, T[$:

The optimal control problem (1.1) is now

$$\min_{u \in \mathcal{U}} \int_{\mathbb{R}} j(X, 0) \rho^0(X) dX : \text{subject to (1.21)} \quad (1.22)$$

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 the open interval $I := (0, X_M)$ the variational

1. As a function of X_t random, j being deterministic has a *PDE* given by Itô calculus, roughly speaking as follows:

$$\begin{aligned} d_t j &= \partial_t j dt + \mathbb{E} [\partial_X j dX] + \frac{1}{2} \mathbb{E} [\partial_{XX} j d\langle X \rangle] \\ &= dt \left(\partial_t j + \partial_X j (r - \kappa X - u) X + \frac{X^2 \sigma^2}{2} \partial_{XX} j \right) = -dt [(X - X_d)^2 + \beta(\partial_t u)^2 - \alpha u]. \end{aligned}$$

formulation of the *PDE* consists in finding j in a weighted Sobolev space:

$$j \in V := \{g \in L^2(\Omega) : X\partial_X g \in L^2(I)\}, \quad j(\cdot, T) = 0 \quad \text{and} \quad \forall \hat{j} \in V, \quad (1.23)$$

$$\begin{aligned} \int_I (\partial_t j + (r - \kappa X - u)X\partial_X j) \hat{j} - \int_I \frac{\sigma^2}{2} \partial_X (X^2 \hat{j}) \partial_X j + \frac{\sigma^2}{2} X_M j(X_M) \hat{j}(X_M) \\ = - \int_I [(X - X_d)^2 + \beta(\partial_t u)^2 - \alpha u] \hat{j}. \end{aligned}$$

1.8.1 Gradient computation

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

$$\begin{aligned} \int_{I \times]0, T[} \text{grad}_u j \delta u := \int_I \rho^0 \delta j(X^0, 0) \quad \text{subject to} \\ \int_I (\partial_t \delta j + (r - \kappa X - u)X\partial_X \delta j) \hat{j} - \int_I \frac{\sigma^2}{2} \partial_X (X^2 \hat{j}) \partial_X \delta j \\ + \frac{\sigma^2}{2} X_M \delta j(X_M) \hat{j}(X_M) = \int_I \hat{j} [(X\partial_X j + \alpha)\delta u - 2\beta\partial_t u \partial_t \delta u], \quad \forall \hat{j} \in V. \end{aligned}$$

Define p by $p(0) = \rho^0$ and

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

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

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

On the other hand,

$$- \int_I \rho^0 \delta j(X, 0) = \int_I \delta j(X, T) p(T) - \int_I \delta j(X, 0) p(0) = \int_{I \times]0, T[} \partial_t (p\delta j).$$

Consequently

$$\int_{I \times]0, T[} \text{grad}_u j \delta u = \int_I \rho^0 \delta j(X^0, 0) = - \int_{I \times]0, T[} [p(X\partial_X j + \alpha)\delta u - 2\beta\partial_t u \partial_t \delta u].$$

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.

1.9 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 (1.22). Let $Y(X, \tau)$ be the solution of

$$\begin{aligned} \dot{Y}(\tau) &= (r - \kappa Y(\tau) - u(Y(\tau), \tau)) Y(\tau), \quad Y(t) = X; \\ \text{then } \partial_t j + (r - \kappa X - u) X \partial_X j &= \frac{d}{d\tau} \rho(Y(\tau, X), \tau)|_{\tau=t}. \end{aligned} \quad (1.24)$$

Hence (1.21) is rewritten as

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

Consequently, when $\sigma \rightarrow 0$, j becomes equal to \tilde{j} , the integral of $-(X - X_d)^2$ on the streamline (1.24) which passes through X^0 at $t = 0$:

$$\begin{aligned} \tilde{j}(Y(T), T) - \tilde{j}(Y(0), 0) &= -j(X^0, 0) = -\int_{Y(0)}^{Y(T)} (X(s) - X_d)^2 \\ \text{i.e. } \tilde{j}(0) &= \int_0^T (X(t) - X_d)^2 dt : \dot{X} = (r - \kappa X - u)X, \quad X(0) = X^0. \end{aligned} \quad (1.25)$$

This analysis shows that *HJB* becomes numerically singular when σ is too small.

1.10 CONCLUSION

Let us compare the 3 classes of methods:

1. Non-dynamic control (NDC) solved by Monte-Carlo.
2. Dynamic Programming in a stochastic setting (DP).
3. Dynamic Programming solved with Partial Differential equations (*HJB*).

NDC is a mathematical oddity because it does not take into account the necessity of a Markovian solution forbidding an attempt to use the current noise level to predict the next step (feedback); but from the practical point of view it is certainly the best solution for the fishing site regulator, especially when it is coupled with Supervised Learning. After all, provided that the model fits the reality, a Markovian solution is not needed. By reinitializing the data every so often at $\{T_k\}_0^J$ and solving J control problem on smaller time intervals $[T_{j-1}, T_j]$, one could borrow from Model Predictive Control (see (5)). Yet, it can be seen on figure 1.4 that the precision is hardly better at early times than at later times.

On this fishing site control problem DP and *HJB* are different implementations of the same approach. All four implementations gave mathematically sound results, but hardly usable because bang-bang, because the Dirac ∂u_t when u is bang-bang, is not handled by the numerical algorithms. Note that a bang-bang saturation of constraints gives a solution which is not far from the elementary but intuitive strategy which consists in taking $u_t = u_m$ when $X_t > X_d$ and u_M otherwise; but we don't want such discontinuous policies. In other words the

penalization of $|\partial_t u|$ works for NDC but not for DP and *HJB* as implemented here. A penalization of $|\partial_x u|$ is probably needed as well.

HJB with *PDEs* is the most taxing numerically and DP is much faster and also capable of generating approximate solution which do not saturate the constraints as much. NDC requires brute force Monte-Carlo, which is also very costly, yet with supervised learning NDC outperform *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 measurements; 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. Finally, porting the trained network to a hand held device is also a major advantage.

ACKNOWLEDGEMENTS

We are thankful to François Murat for his help to minimize the hypotheses necessary for the existence of a solution to the Kolmogorov control problem.

The computer programs in Python and FreeFEM++ can be obtained by email from olivier.pironneau@sorbonne-universite.fr.

BIBLIOGRAPHY

- [1] 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.
- [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] R. Bellman, *Dynamic Programming*, Princeton, NJ: Princeton University Press, 1957.
- [4] D. Bertsekas, *Reinforced Learning & Optimal Control*. Athena Scientific, Belmont Mass. 2019.
- [5] S. Boyd and C. Barratt: *Linear Controller Design*. Prentice-Hall, 1991.
- [6] 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.
- [7] F. Chollet: *Deep learning with Python*. Manning publications (2017).
- [8] I. Goodfellow, Y. Bengio and A. Courville (2016): *Deep Learning*, MIT-Bradford.
- [9] F. Hecht (2012): New development in FreeFem++, *J. Numer. Math.*, 20, pp. 251-265. (see also www.freefem.org.)
- [10] Jiongmin Yong and Xun Yu Zhou: *Stochastic Controls Hamiltonian Systems and HJB Equations* Application of Mathematics series vol 43. Springer 1991.
- [11] R. Kamalapurkar and P. Walters and J. Rosenfeld and W. Dixon, *Reinforcement Learning for Optimal Feedback Control*. Springer 2018.
- [12] M. Lauriere and O. Pironneau: Dynamic Programming for mean-field type control *J. Optim. Theory Appl.* 169 (2016), no. 3, 902–924.

- [13] C. Le Bris and P.L. Lions, Existence and uniqueness of solutions to Fokker-Planck type eqs. with irregular coefficients. *Comm. Partial Differential Eqs.*, 33, 1272-1317, 2008.
- [14] G. Pagès, H. Pham and J. Printems: An Optimal Markovian Quantization Algorithm For Multi-Dim. Stochastic Control Problems, *Stochastics and Dynamics*, 4(4):501–545, 2004.
- [15] G. Pagès. *Numerical Probability: An Introduction with Applications to Finance*. Springer, Berlin, 2018, 574p.
- [16] G. Pagès, J. Printems. Optimal quadratic quantization for numerics: the Gaussian case, *Monte Carlo Methods and Appl.*, 9(2):135–165, 2003.

APPENDIX A

1.11 A MODEL WITH FISHING QUOTA

Let $X(t)$ be the biomass at time t , $E(t)$ the fishing effort – interpreted as the number of boats – and $Q(t)$ is the fishing capacity per boat. In (6) $Q(t) = qX(t)$, 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:

$$\dot{X}(t) = X(t)(r - \kappa'X(t)) - \min(qX(t), Q(t))E(t). \quad (\text{Biomass dynamics.})$$

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.

$$\dot{E} = E(pF - c) \quad (\text{Fishing effort driven by profit,})$$

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

$$\Phi \dot{p} = D(p) - FE \quad \text{with the demand } D(p) = \frac{a'}{1 + \gamma p}$$

where Φ is the inverse time scale at which the fish market price adjusts. When $\Phi \ll 1$, it may be approximated by $D(p) = FE$ i.e. $pFE = \frac{a'}{\gamma} - \frac{FE}{\gamma}$. With $a = a'/\gamma$ and $\kappa = \kappa'\gamma$, the whole system is:

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

Now notice that by changing $X_0 \rightarrow X_0/\gamma$ we may take $\gamma = 1$.

Remark 4. 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 5. The ODE system is of the form $\dot{Y}(t) = \psi(Y(t))$, $t \in]0, T[$, $\psi : \mathbb{R}^2 \rightarrow \mathbb{R}^2$. 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 1.18 with the following parameters (see file `fishingQuota0.py` and for the definition of α and β , see below):

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

Notice that q is big enough so that $\min(qX(t), Q(t)) = Q(t)$, $\forall t$. Notice also that without quota the biomass tends to zero, in this case.

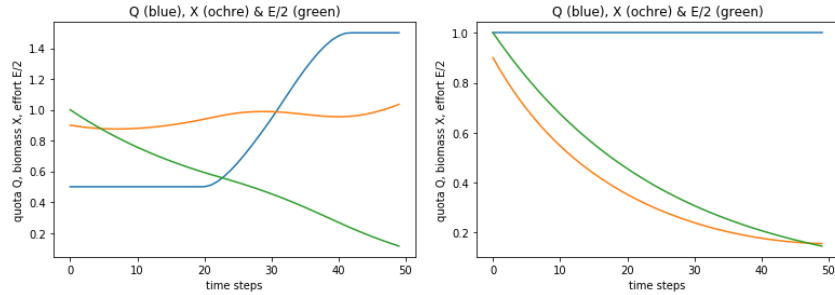


FIGURE 1.18 Left: Quota (blue) and fishing effort (ochre). No quota means that $Q(t) = 1$ for all t . Right: biomass with (blue) and without quota (ochre).

1.12 REFORMULATION

To prevent 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 quotas are irrelevant in the sense that the fisherman is given 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

$$\min_{Q' \in Q'} \{ \bar{J} := \int_0^T (X(t) - X_d(t))^2 dt : \dot{X} = X(r - \kappa'X - u(t)), X(0) = X^0 \}$$

In the paper, the optimization is performed without computing E ; then E is computed with the optimal u by solving $\dot{E} = a - cE - uX$, $E(0) = E^0$. As said before it is not feasible to impose too 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.

1.12.1 More constraints on quotas

Low quotas are unpopular. To avoid them we may use penalty and add to the criteria $-\int_0^T \alpha u(t) dt$.

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 criteria a term like $\beta \left| \frac{du}{dt} \right|^2$.

An optimal policy is a solution of

$$\min_{Q \in \mathcal{Q}} J(u) := \int_0^T \left[|X - X_d|^2 - \alpha u + \beta \left| \frac{du}{dt} \right|^2 \right] : \text{subject to}$$

$$\dot{X} = X(r - \kappa'X - u(t)), X(0) = X^0.$$

Figure 1.1 shows the solution with the same parameters as for figure 1.18. 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 one.

1.13 AN ANALYTICAL SOLUTION FOR A SIMILAR PROBLEM

It was said earlier that $Y_t = \log X_t$ simplify the problem. If the goal $X_t \sim X_d$, is changed to $Y_t \sim Y_d$, then with $v_t = r - \frac{\sigma^2}{2} - \kappa e^{Y_t} - u$ the problem is

$$\begin{aligned} \min_{u \in \mathcal{U}} \left\{ \int_0^T \mathbb{E} [(Y_t - Y_d)^2 - \alpha u + \beta |\partial_t u|^2] \text{ with } dY_t = v_t(u) dt + \sigma dW_t, \right\} \\ \Rightarrow 0 = \inf_{u \in \mathcal{U}} \left\{ \partial_t V + v \partial_Y V + \frac{\sigma^2}{2} \partial_{YY} V + (Y - Y_d)^2 - \alpha u + \beta |\partial_t u|^2 \right\}. \end{aligned}$$

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 $a(T) = b(T) = c(T) = \dot{K}(T) = \dot{k}(T) = 0$. By identification for all Y it is found that:

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

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:

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

Results, obtained with finite differences, are shown on figures 1.19 & 1.20.

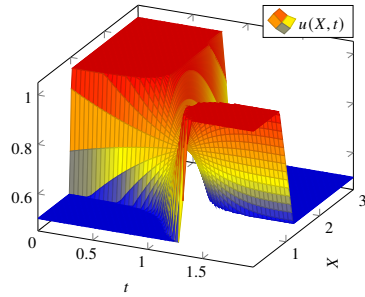


FIGURE 1.19 Solution of the Dynamic when $X_0 = 0.8$ and comparison with the Programming equation: side view.

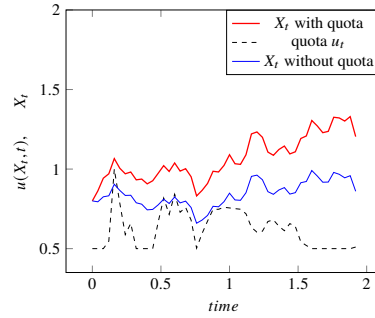


FIGURE 1.20 Optimal biomass and quota when $X_0 = 0.8$ and comparison with the solution with $u = u_M$.