# Delayed Investment Decisions in Renewable Energy under Uncertainty: A Deep Learning–Based Approach

Insaf Agram[1], Abdelhak Rais[1] and Nacira Agram[2]
[1] Department of Economic Sciences, University of Biskra, Algeria
[2] Department of Mathematics, KTH Royal Institute of Technology, Stockholm, Sweden
`insaf.agram@univ-biskra.dz`

**Abstract.** We investigate a stochastic control problem for renewable energy capacity installation under uncertainty and implementation delay. Investment decisions are irreversible and subject to time-to-build constraints such as construction, regulatory approval, and grid integration.

Electricity demand uncertainty and renewable intermittency are modeled through jump-driven stochastic dynamics, capturing both continuous fluctuations and rare extreme events. The introduction of delay induces path dependence and leads to a non-Markovian control problem.

To address this challenge, we propose a deep learning-based global control framework that directly approximates optimal feedback policies from simulated trajectories. Unlike dynamic programming or BSDE-based methods, the approach avoids value function approximation and remains tractable in high-dimensional and delayed settings.

Numerical experiments show that delay significantly alters optimal investment timing and induces smoother, anticipative strategies.

**Keywords:** Delay system, renewable energy, deep learning

## 1 Introduction

Renewable energy investment decisions are characterized by irreversibility, uncertainty, and long implementation horizons. Electricity demand is inherently stochastic, renewable production is intermittent, and energy markets are subject to regulatory and geopolitical shocks.

A key aspect of real-world investment processes is the presence of implementation delays. Installation decisions involve construction, permitting, and grid integration, implying that control actions only take effect after a time lag.

From a mathematical perspective, the introduction of delay fundamentally alters the structure of the control problem. The system becomes path-dependent and loses the Markov property, making classical dynamic programming approaches inapplicable.

## 2 Delayed Renewable Installation Model

State variable:

D. Agti et al. (eds.), *Proceedings of the International Conference on Artificial Intelligence Applications in Business Administration in MENA Region (ICAIABA 2026)*, Advances in Economics, Business and Management Research 393,
https://doi.org/10.2991/978-94-6239-711-8_40

$X(t) = (V(t), D(t), C_R^{A,\delta}(t))$
Dynamics:
$dV(t) = b_V(V(t))\,dt + \int \beta_V(V(t-), z)\,\tilde{N}(dt,dz)$
$dD(t) = b_D(D(t))\,dt + \int \beta_D(D(t-), z)\,\tilde{N}(dt,dz)$
Delayed dynamics:
$dC_R^{A,\delta}(t) = \int (V(t-\delta) - A(t-\delta))^+ \odot \beta_V(\ldots)\,\tilde{N}(dt,dz)$
Cost functional:
$J^\delta(A) = E[\int e^{-rt}(D(t) - V(t)^T C_R(t))^+\,dt + \kappa^T C_R(T)]$

## 3 Deep Control Approach with Delay

Control parameterization:
$A_\theta(t,x) = \mathcal{A}_\theta(t, X(t))$
Discretization:
$k = \mathrm{floor}(\delta / \Delta t)$

**Algorithm: Delayed Deep Control Solver**

```
Input: Delay $\delta$, time grid $\{t_n\}$, batch size B
Initialize neural network parameters $\theta$
For each training epoch:
    For j = 1,...,B:
        Initialize $X_0^j$
    For n = 0,...,M-1:
        For j = 1,...,B:
            Compute control:
                $A_n^j = \mathcal{A}_\theta(t_n, X_n^j)$
            If a jump occurs:
                Apply delayed control using index (n - k)
            Update system:
                $V_n^j$, $D_n^j$ using jump dynamics
    Compute loss based on cost functional $J^\delta$
    Update $\theta$ using gradient descent (Adam optimizer)
Output: Optimal parameter $\theta^*$ and feedback control $A_\theta$
```

## 4 Numerical Results

Neural network with two hidden layers (256 neurons), trained with Adam (learning rate 1e-4).

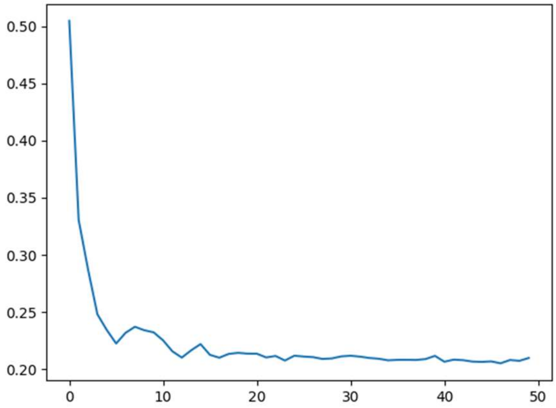

Figure 1: Convergence of loss

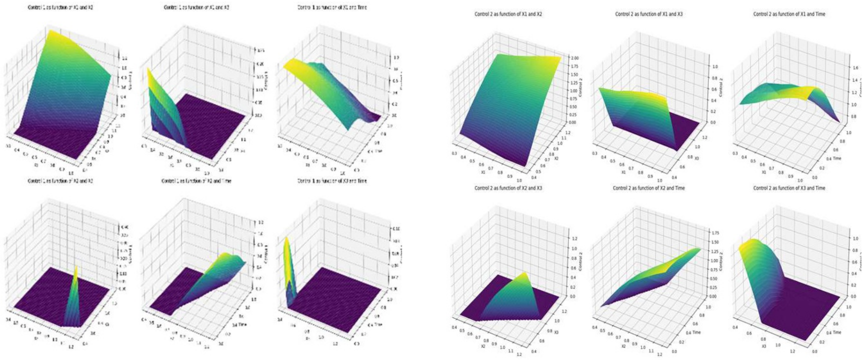

Figure 2: Control surface

Results show delay induces smoother and anticipative investment strategies.

## 5 Conclusion:

The neural network consists of an input layer, two hidden layers with 256 neurons each, and an output layer. ReLU activation functions are used in the hidden layers, and parameters are initialized using the Kaiming normal initialization. Training is performed using the Adam optimizer with learning rate 10−4 and batch size 2000. The time horizon is discretized using M = 50 time steps, and the network is trained for 50 epochs.

The results highlight the significant impact of implementation delays on optimal capacity installation strategies. This observation is consistent with the numerical findings in (?), where optimal installation is shown to occur at a single jump event under jump-driven uncertainty.

## References

1. Yong, J., Zhou, X.Y. (1999). Stochastic Controls: Hamiltonian Systems and HJB Equations. Springer.
2. Pham, H. (2009). Continuous-time Stochastic Control and Optimization. Springer.
3. Peng, S. (1992). Dynamic programming principle. Stochastics Reports.
4. Benth, F.E. et al. (2014). Electricity Markets Modeling. World Scientific.
5. Carmona, R., Ludkovski, M. (2010). Energy storage valuation. Quantitative Finance.
6. Dixit, A., Pindyck, R. (1994). Investment Under Uncertainty. Princeton.
7. Han, J. et al. (2018). Deep learning PDE solver. PNAS.
8. Pham, H., Warin, X. (2019). Neural PDE methods.