Auto-Encoding Risk-Neutral Model — Step-by-Step Explanation

Gary Pai

November 2025

1 Auto-Encoding Risk-Neutral Model (Lyashenko–Mercurio–Sokol, 2024)

This section explains the key ideas of the auto-encoding risk-neutral model step-by-step, including a simple numerical illustration.

1.1 Step 1: Representing the Yield or Forward Curve

Let $f(\tau)$ denote the instantaneous forward rate curve as a function of maturity τ. For a given day, market quotes provide discretely sampled forward rates:

$$f(\tau_1), f(\tau_2), \ldots, f(\tau_N).$$

These points form a vector in R^N. We feed this vector into an autoencoder.

1.2 Step 2: Passing the Curve into an Autoencoder

An autoencoder consists of:

- an encoder $E(\cdot)$,
- a decoder $D(\cdot)$,
- a low-dimensional latent state $z \in R^d$.

The encoder maps the full forward curve into a latent vector:

$$z = E(f).$$

The decoder reconstructs the curve:

$$\hat{f} = D(z).$$

The latent dimension d is typically small, e.g. $d = 2$ or $d = 3$.

1.3 Step 3: The Latent Space is Treated as the Risk-Neutral State Variable

The crucial idea of the model is:

$$z_t becomes the state variable driving risk-neutral dynamics.$$

Under the risk-neutral measure Q:

$$dz_t = \mu(z_t)\,dt + \sigma(z_t)\,dW_t^Q.$$

The forward curve is then reconstructed from:

$$f_t(\tau) = D(z_t)(\tau).$$

Thus, pricing depends on a low-dimensional diffusion instead of a full infinite-dimensional curve.

1.4 Step 4: No-Arbitrage Constraints Are Enforced via the Decoder

Instead of imposing no-arbitrage conditions on $f(\tau)$ directly, the paper imposes constraints on the decoder:

$$f(\tau) = D(z)(\tau) \quad \Rightarrow \quad no-arbitrage shape conditions.$$

Examples enforced on $D(z)$:

- monotonic bond price condition $P(\tau)$,
- smoothness of $f(\tau)$,
- positivity of forward rates (optional),
- correct asymptotic long-term rate $f(\infty)$.

Thus all decoded curves are guaranteed to lie on an arbitrage-free manifold.

1.5 Step 5: Learning the Model from Data

The training objective is:

$$\mathcal{L} = \underbrace{\|f - D(E(f))\|^2}_{reconstruction loss} + \underbrace{\lambda_{shape} \cdot \text{ShapePenalty}(D(z))}_{no-arbitrage} + (regularization terms).$$

After training:

- The encoder finds optimal latent coordinates z for each curve.
- The decoder learns a smooth, arbitrage-free curve surface.

1.6 Step 6: A Simple Numerical Example

Suppose the forward curve has quotes at maturities:

$$\tau = (1, 2, 5, 10) years,$$

with observed forwards (in %):

$$f = (4.5,\ 4.2,\ 3.9,\ 3.8).$$

Encoding

Assume the trained encoder produces:

$$z = E(f) = (0.80,\ -0.25).$$

Decoding

The decoder reconstructs:

$$D(z) = (4.48,\ 4.18,\ 3.92,\ 3.79),$$

which is close to the observed curve and satisfies the imposed shape constraints.

1.7 Step 7: Risk-Neutral Evolution in Latent Space

Under the risk-neutral measure:

$$dz_t = (\ \mu\ )_1\ (z_t)\mu_2(z_t)dt + (\ \sigma\ )_{11}\ (z_t)\sigma_{12}(z_t)\sigma_{21}(z_t)\sigma_{22}(z_t)dW_t^Q.$$

For example, a simple Ornstein–Uhlenbeck form:

$$dz_t = -A(z_t - \theta)\, dt + \Sigma\, dW_t^Q.$$

The forward curve at time t is:

$$f_t(\tau) = D(z_t)(\tau).$$

1.8 Step 8: Why the Model is Powerful

- Only d dimensions (e.g., 2–3) drive the entire yield curve.
- All curves are guaranteed to be arbitrage-free.
- The model is data-driven but consistent with finance theory.
- The state process z_t is low-dimensional and easy to simulate.

1.9 Summary

The Lyashenko–Mercurio–Sokol model replaces traditional factor models with a learned arbitrage-free manifold of yield curves. The autoencoder extracts latent coordinates z_t, which evolve under Q according to a diffusion process. The decoder reconstructs yield curves from these coordinates, ensuring that all simulated and fitted curves remain arbitrage-free.