Modeling Macro-Economic Scenarios: An Ornstein-Uhlenbeck Approach

Niels van Winden
Wednesday, January 29, 2025
Version 1.0

Abstract. The Ornstein-Uhlenbeck (OU) process is a widely used continuous-time stochastic process that serves as the continuous analog of an AR(1) model [6]. In this work, we extend this framework to the second-order Ornstein-Uhlenbeck (OU(2)) process, which incorporates both mean-reverting and cyclical dynamics, making it well-suited for macro-economic scenario generation. We derive its mathematical properties, demonstrate its equivalence to an AR(2) process under Euler-Maruyama discretization [1, 4], and apply it to economic indicators such as unemployment rates and house price indices. Model parameters are estimated using maximum likelihood and Kalman filtering, with model fit evaluated via AIC and BIC criteria. Our results indicate that the OU(2) model outperforms standard AR(2) models in capturing both short-term fluctuations and long-term trends, confirming its effectiveness in economic scenario modeling.

Contents

1 Introduction

The Ornstein-Uhlenbeck (OU) process, introduced by Ornstein and Uhlenbeck [6], was originally developed to model Brownian motion with friction. Over time, it has found widespread applications in physics, finance, and economics due to its properties of stationarity, Markovian structure, Gaussian distribution, and mean reversion. It has been extensively used in financial modeling, including term structure models such as the Vasicek interest rate model [7], as well as broader applications in stochastic modeling [2].

Since economic time series often exhibit cyclical behavior beyond simple mean reversion, we extend the classical OU framework to a second-order Ornstein-Uhlenbeck (OU(2)) process. This model introduces oscillatory components, allowing it to better capture macroeconomic indicators that experience periodic fluctuations. We derive its mathematical formulation, analyze its autocorrelation structure, and establish its equivalence to an AR(2) model via Euler-Maruyama discretization [4].

To validate the OU(2) model, we apply it to macro-economic data such as unemployment rates and house price indices. Using maximum likelihood estimation and Kalman filtering, we estimate model parameters and assess fit using AIC and BIC criteria. Our findings suggest that the OU(2) process offers a superior fit for high-frequency economic data, effectively capturing both transient fluctuations and long-term trends. This study demonstrates how continuous-time stochastic models can enhance macro-economic scenario generation, providing an alternative to traditional discrete-time approaches.

2 Introduction to the Ornstein-Uhlenbeck Process

The Ornstein-Uhlenbeck (OU) process is a well-known continuous-time stochastic process that plays a significant role in various fields such as physics, finance, and biology. It is a mean-reverting process, meaning that the system has a tendency to drift back towards its long-term mean over time. This makes it particularly useful for modeling systems where fluctuations are centered around a stable equilibrium.

2.1 Mathematical Formulation

The OU process can be defined by the stochastic differential equation [1]:

$$dX_t = \theta(\mu - X_t)dt + \sigma dW_t \tag{2.1.1}$$

where:

- X_t is the state of the process at time t,
- μ is the long-term mean towards which the process reverts,
- $\theta > 0$ is the rate of mean reversion, controlling how quickly the process returns to μ,
- $\sigma > 0$ represents the volatility or the intensity of the random fluctuations,
- W_t is a standard Wiener process (or Brownian motion) at time t.

The OU process has several important characteristics [1]:

1. **Stationarity:** The process has a stationary distribution when $\theta > 0$, given by a normal distribution with mean μ and variance $\frac{\sigma^2}{2\theta}$.
2. **Mean Reversion:** Deviations from the mean μ are corrected over time at a rate determined by θ.
3. **Autocorrelation:** The correlation between X_t and X_s decays exponentially with $|t-s|$, as described by $\text{Corr}(X_t, X_s) = e^{-\theta|t-s|}$.

The combination of mean-reverting behavior and random fluctuations makes the OU process a cornerstone in stochastic modeling. Subsequent sections will explore an extended version and applications in macro-economic scenario generation.

2.2 Equivalence of AR(1) Process under Euler-Maruyama Discretization

Now notice that the Ornstein-Uhlenbeck process defined in equation (2.1.1) is first order. Therefore we can utilize the first order accurate Euler-Maruyama method to approximate its solution, thereby obtaining:

$$X_{t+\Delta t} = X_t + \theta(\mu - X_t)\Delta t + \sigma\sqrt{\Delta t}\epsilon_t \tag{2.2.1}$$

where $\epsilon_t \sim \mathcal{N}(0,1)$ represent independent standard normal random variables. Rearranging terms gives:

$$X_{t+\Delta t} = (1 - \theta\Delta t)X_t + \theta\mu\Delta t + \sigma\sqrt{\Delta t}\epsilon_t$$

This equation resembles the form of an autoregressive process of order 1 (AR(1)):

$$X_{t+1} = \phi X_t + c + \eta_t \tag{2.2.2}$$

Under the transformation:

- $\phi = 1 - \theta\Delta t$ is the lag coefficient,
- $c = \theta\mu\Delta t$ is the intercept,
- $\eta_t = \sigma\sqrt{\Delta t}\epsilon_t$ is the noise term with variance $\sigma^2\Delta t$.

Thus, under the Euler-Maruyama approximation, the AR(1) process can be viewed as a discretised OU process, with parameters depending on the time step Δt and the model coefficients θ, μ, and σ.

In practical applications, data is often collected at discrete time intervals, making it challenging to directly estimate the parameters of a continuous-time stochastic process like the OU process. However, by leveraging the equivalence to an AR(1) process, one can approximate the parameters of the OU process using the above parameter transformation whilst also leveraging the known theoretical characteristics of the first order OU process.

3 Second Order Ornstein-Uhlenbeck Process

The Second-Order Ornstein-Uhlenbeck (OU) process is an extension of the classical first-order OU process. The process is defined by the stochastic differential equation (SDE):

$$\ddot{X}(t) + \gamma \dot{X}(t) + \omega^2 X_t = \sigma \dot{W}_t \tag{3.0.1}$$

where γ is the damping coefficient, ω is the natural frequency of the system, σ is the noise intensity, and $\dot{W}_t$ is Gaussian white noise. The double dots represent the second time derivative.

3.1 Homogeneous Solution

The homogeneous solution reveals the system's intrinsic dynamics in the absence of external noise, highlighting its stability, damping properties, and natural frequencies. It characterizes how deviations from equilibrium evolve over time, whether they decay exponentially or oscillate before settling. To find the homogeneous solution, we set the right-hand side of equation (3.0.1) to zero:

$$\ddot{X}(t) + \gamma \dot{X}(t) + \omega^2 X(t) = 0 \tag{3.1.1}$$

Assuming a solution of the form $X(t) = e^{rt}$, we obtain the characteristic equation:

$$r^2 + \gamma r + \omega^2 = 0$$

Solving for $r_{1,2}$:

$$r = \frac{-\gamma \pm \sqrt{\gamma^2 - 4\omega^2}}{2}$$

The general solution depends on the nature of the roots: If $\gamma^2 > 4\omega^2$, the roots are real and distinct, then:

$$X_h(t) = C_1 e^{r_1 t} + C_2 e^{r_2 t}$$

If $\gamma^2 = 4\omega^2$, there is a repeated root $r = -\gamma/2$, so:

$$X_h(t) = (C_1 + C_2 t) e^{-\gamma t/2}$$

If $\gamma^2 < 4\omega^2$, the roots are complex:

$$r = \frac{-\gamma}{2} \pm i \frac{\sqrt{4\omega^2 - \gamma^2}}{2}$$

Defining $\nu = \frac{\sqrt{4\omega^2 - \gamma^2}}{2}$, the solution becomes:

$$X_h(t) = e^{-\gamma t/2} \left(C_1 \cos(\nu t) + C_2 \sin(\nu t) \right) \tag{3.1.2}$$

Thus, this gives the homogeneous solution $X_h(t)$. We will use this solution for the derivation of the mean reversion in section §4.3.

3.2 A State Space Representation of the OU(2) Process

To analyze the second-order Ornstein-Uhlenbeck (OU(2)) process in a structured manner, we introduce a state-space representation. The state-space form expresses the system as a set of first-order differential equations, which is particularly useful for numerical simulations and control applications. Define the state vector:

$$\mathbf{X}(t) = \begin{bmatrix} X_t \\ \dot{X}(t) \end{bmatrix}$$

where X_t is the position variable and $\dot{X}(t)$ is the velocity. Additionally, define the state matrix:

$$A = \begin{bmatrix} 0 & 1 \\ -\omega^2 & -\gamma \end{bmatrix}$$

And lastly, define the volatility matrix:

$$B = \begin{bmatrix} 0 & 0 \\ 0 & \sigma \end{bmatrix}$$

Then rewriting the second-order stochastic differential equation (3.0.1) in terms of the state vector yields the first-order system:

$$\dot{\mathbf{X}}(t) = A\mathbf{X}(t) + B\dot{W}_t \tag{3.2.1}$$

which defines the deterministic part of the system dynamics. which scales the effect of the Gaussian white noise input $\dot{W}_t$. The state-space representation is valuable because it allows for straightforward numerical integration and filtering techniques such as the Kalman filter to estimate system states based on noisy observations. We will use these advantages in section §5.3 to estimate our parameters. The eigenvalues of the matrix A determine the stability properties of the process, with damping effects introduced by γ ensuring mean reversion.

3.3 Discrete State-Space Representation

To analyze a second-order Ornstein-Uhlenbeck (OU) process numerically, it is often necessary to discretize its continuous-time state-space representation. The goal is to obtain an equivalent discrete-time system of equation (3.0.1). Using the matrix exponential, the discrete-time equivalent system with time step Δt is given by:

$$\mathbf{x}_{k+1} = e^{A\Delta t}\mathbf{x}_k + \int_0^{\Delta t} e^{A(\Delta t - s)} B d\mathbf{w}(s) \tag{3.3.1}$$

Defining $A_d = e^{A\Delta t}$ and using the properties of stochastic integrals, the discrete-time system can be expressed as:

$$\mathbf{x}_{k+1} = A_d\mathbf{x}_k + \mathbf{q}_k \tag{3.3.2}$$

where $\mathbf{q}_k$ is a Gaussian noise term with mean zero and covariance matrix:

$$Q_d = \int_0^{\Delta t} e^{As} B B^T e^{A^T s} ds \tag{3.3.3}$$

This formulation allows numerical simulation and estimation techniques, such as the Kalman filter, to be applied efficiently while preserving the statistical properties of the continuous process.

3.4 Discrete Euler-Maruyama Representation

To convert equation (3.0.1) into a first-order system, we define the state variables:

$$X_1(t) = X(t), \quad X_2(t) = \dot{X}(t)$$

Rewriting the system in terms of these variables, we obtain:

$$\dot{X}_1 = X_2$$

$$\dot{X}_2 = -\gamma X_2 - \omega^2 X_1 + \sigma \dot{W}(t)$$

3.4.1 Euler-Maruyama and Milstein Discretization Equivalence

Applying the Euler-Maruyama scheme[1] to our system, we get:

$$X_1^{n+1} = X_1^n + X_2^n \Delta t$$

$$X_2^{n+1} = X_2^n + (-\gamma X_2^n - \omega^2 X_1^n)\Delta t + \sigma \Delta W_n$$

Since the noise term is additive, both Euler-Maruyama and Milstein methods yield the same result in this case.

[1] `https://en.wikipedia.org/wiki/Euler-Maruyama_method`

3.4.2 Equivalence to an AR(2) Process under Euler-Maruyama Discretization

Discretizing the system using the Euler-Maruyama method with step size Δt, we approximate:

$$X_{t+\Delta t} = X_t + \Delta t Y_t \tag{3.4.2.1}$$

$$Y_{t+\Delta t} = Y_t - \Delta t(\gamma Y_t + \omega^2 X_t) + \sigma\sqrt{\Delta t}\xi_t \tag{3.4.2.2}$$

where $\xi_t \sim \mathcal{N}(0,1)$ represents standard Gaussian noise, and $Y_t = \dot{X}(t)$ denotes the velocity.

To eliminate Y_t, we substitute (3.4.2.1) into (3.4.2.2) and shift indices accordingly:

$$Y_{t+\Delta t} = \frac{X_{t+\Delta t} - X_t}{\Delta t}$$

Substituting this into (3.4.2.2), we obtain:

$$\frac{X_{t+2\Delta t} - X_{t+\Delta t}}{\Delta t} = \frac{X_{t+\Delta t} - X_t}{\Delta t} - \Delta t\left(\gamma\frac{X_{t+\Delta t} - X_t}{\Delta t} + \omega^2 X_{t+\Delta t}\right) + \sigma\sqrt{\Delta t}\xi_t$$

Rearranging terms:

$$X_{t+2\Delta t} - 2X_{t+\Delta t} + X_t = -\gamma\Delta t(X_{t+\Delta t} - X_t) - \omega^2\Delta t^2 X_{t+\Delta t} + \sigma\sqrt{\Delta t}\xi_t$$

Recognizing this as an AR(2) process:

$$X_{t+2\Delta t} = a_1 X_{t+\Delta t} + a_2 X_t + \sigma\sqrt{\Delta t}\xi_t$$

we identify the AR(2) coefficients:

$$a_1 = 2 - \gamma\Delta t - \omega^2\Delta t^2 \tag{3.4.2.3}$$

$$a_2 = -(1 - \gamma\Delta t) \tag{3.4.2.4}$$

Thus, the second-order Ornstein-Uhlenbeck process, when discretized using Euler-Maruyama, follows an AR(2) structure. The AR(2) parameters depend explicitly on γ, ω, and the discretization step Δt. The equivalence holds in an approximate sense, with higher-order terms omitted, making the relationship exact only in the limit $\Delta t \to 0$.

3.5 Frequency Domain Representation of OU(2) Processes

To analyze the system in the frequency domain, we apply the Fourier transform to equation (3.0.1). The Fourier transform of a time-domain function $f(t)$ is defined as:

$$\tilde{f}(\nu) = \int_{-\infty}^{\infty} f(t) e^{-i\nu t}\, dt$$

where ν is the angular frequency variable, and $\tilde{f}(\nu)$ is the Fourier transform of $f(t)$. Applying the Fourier Transform to each term in (3.0.1) gives rise to the equations:

$$\mathcal{F}[X_t] = \tilde{X}(\nu)$$

$$\mathcal{F}[\dot{X}_t] = i\nu\tilde{X}(\nu)$$

$$\mathcal{F}[\ddot{X}_t] = -\nu^2\tilde{X}(\nu)$$

$$\mathcal{F}[\dot{W}_t] = \tilde{\xi}(\nu)$$

(For more detail on this derivation see §7.1)

Substituting these transformed terms into the equation of motion yields:

$$(-\nu^2)\tilde{X}(\nu) + \gamma(i\nu)\tilde{X}(\nu) + \omega^2\tilde{X}(\nu) = \sigma\tilde{\xi}(\nu)$$

Factoring out $\tilde{X}(\nu)$ on the left-hand side:

$$\left(-\nu^2 + i\gamma\nu + \omega^2\right)\tilde{X}(\nu) = \sigma\tilde{\xi}(\nu)$$

Then the frequency-domain representation of the equation of motion is denoted:

$$\tilde{X}(\nu) = \frac{\sigma\tilde{\xi}(\nu)}{-\nu^2 + i\gamma\nu + \omega^2} \tag{3.5.1}$$

The factor $-\nu^2 + i\gamma\nu + \omega^2$ represents the system's frequency response or transfer function, and $\tilde{\xi}(\nu)$ represents the Fourier transform of the noise input. The assumption of stationarity implies that the statistical properties of X_t, such as its mean, variance, and autocorrelation, are time-invariant. This allows us to analyze the system in terms of its spectral properties, focusing on the distribution of amplitude or power across different frequency components. We will use the above frequency-domain representation to derive the Auto Correlative Function (ACF) of OU(2) processes in section §4.1.

4 Empirical Properties of OU(2) Processes

4.1 Autocorrelation Function (ACF)

The autocorrelation function R_X of the Second-Order OU process can be derived as follows. Starting with the governing equation:

$$\ddot{X}(t) + \gamma\dot{X}(t) + \omega^2 X_t = \sigma W_t \tag{4.1.1}$$

where we assume X_t to be stationary and take the Fourier transform of its governing equation. Let $\tilde{X}(\nu)$ and $\tilde{\xi}(\nu)$ represent the Fourier transforms of X_t and W_t, respectively. Then as discussed in §3.2:

$$\tilde{X}(\nu) = \frac{\tilde{\xi}(\nu)}{-\nu^2 + i\gamma\nu + \omega^2}$$

Then power spectral density (PSD) S_X of the process is given by:

$$S_X(\nu) = \frac{\sigma^2}{(\omega^2 - \nu^2)^2 + \gamma^2\nu^2}$$

(For more detail see appendix §7.1.1)

By the Wiener-Khinchin theorem[2], the autocorrelation function R_X is the inverse Fourier transform of the PSD:

$$R_X(\tau) = \int_{-\infty}^{\infty} S_X(\nu) e^{i2\pi\nu\tau} \mathrm{d}\nu$$

To evaluate this integral, we rewrite the denominator in the PSD in terms of the damping ratio $\zeta = \frac{\gamma}{2\omega}$ and natural frequency ω:

$$S_X(\nu) = \frac{\sigma^2}{(\omega^2 - \nu^2)^2 + \gamma^2\nu^2} = \frac{\sigma^2}{\omega^4(1 - (\nu/\omega)^2)^2 + 4\zeta^2\omega^2\nu^2}$$

Using contour integration or standard techniques for solving inverse Fourier transforms, the resulting autocorrelation function is:

$$R_X(\tau) = \frac{\sigma^2}{2\omega\sqrt{1-\zeta^2}} e^{-\zeta\omega|\tau|} \cos\left(\omega\sqrt{1-\zeta^2}|\tau|\right) \tag{4.1.2}$$

(See appendix §7.1.2 for further details)

Here, the exponential decay term $e^{-\zeta\omega|\tau|}$ reflects the damping effect, while the cosine term $\cos\left(\omega\sqrt{1-\zeta^2}|\tau|\right)$ introduces oscillatory behaviour due to the natural frequency of the system. One point to note is that this formula considers undamped case i.e. $0 \leq \zeta < 1$.

[2] `https://mathworld.wolfram.com/Wiener-KhinchinTheorem.html`

4.2 Variance

The variance of the process, $\mathrm{Var}(X_t)$, is obtained as the value of the autocorrelation function at $\tau = 0$:

$$\mathrm{Var}(X_t) = \frac{\sigma^2}{2\omega\sqrt{1-\zeta^2}} \tag{4.2.1}$$

This result is consistent with the stationary solution of the second-order stochastic differential equation. This represents the steady-state variance of the system under stochastic forcing.

4.3 Mean Reversion

Mean reversion describes the tendency of the process to return to its mean over time. For the Second-Order OU process, the mean-reversion time scale τ_r is related to the damping ratio ζ and the natural frequency ω present in the homogeneous solution (**??**). Specifically, the exponential term $e^{-\zeta\omega t}$ in (**??**) governs the decay of perturbations. The characteristic timescale for mean reversion is determined by the decay rate of this term, given by:

$$\tau_r = \frac{1}{\text{decay rate}} = \frac{1}{\zeta\omega}$$

Thus, the mean reversion timescale is:

$$\tau_r = \frac{1}{\zeta\omega} \tag{4.3.1}$$

The smaller the damping ratio, the slower the mean reversion, whereas higher damping results in faster stabilization around the mean.

4.4 Expectation Value

Unlike in Section §2.1, where the process reverts back to μ, the expectation value $\mathbb{E}[X_t]$ of the Second-Order OU process is zero for a stationary process. The general solution of the Second-Order OU process consists of a homogeneous solution $X_h(t)$ and an inhomogeneous solution $X_p(t)$, given by:

$$X(t) = X_h(t) + X_p(t)$$

From the homogeneous solution (**??**), we have:

$$X_h(t) = e^{-\zeta\omega t}\left(C_1\cos(\omega\sqrt{1-\zeta^2}t) + C_2\sin(\omega\sqrt{1-\zeta^2}t)\right)$$

Taking the expectation on both sides, we obtain:

$$\mathbb{E}[X_h(t)] = e^{-\zeta\omega t}\left(C_1\mathbb{E}[\cos(\omega\sqrt{1-\zeta^2}t)] + C_2\mathbb{E}[\sin(\omega\sqrt{1-\zeta^2}t)\right)]$$

Due to linearity of the expectation operator $\mathbb{E}$ and the deterministic nature of $e^{-\zeta\omega t}$. Furthermore, since the expectation values of the sine and cosine terms vanish in a stationary process, and the exponential decay ensures any transient effects disappear over time, we conclude:

$$\mathbb{E}[X_h(t)] = 0$$

For the inhomogeneous solution $X_p(t)$, in the presence of a stationary driving noise, the system reaches an equilibrium where the long-term expectation value remains constant. Since the process is centered around zero in equilibrium, we have:

$$\mathbb{E}[X_p(t)] = 0$$

Thus, taking the expectation of the full solution:

$$\mathbb{E}[X_t] = \mathbb{E}[X_h(t)] + \mathbb{E}[X_p(t)] = 0 \tag{4.4.1}$$

Therefore, for a stationary Second-Order OU process, the expectation value is zero, meaning that the process fluctuates symmetrically around zero. If the process starts with a nonzero initial condition, these transient effects decay at a rate $\zeta\omega$, leaving a stationary process with zero mean.

In Summary
The second-order Ornstein-Uhlenbeck process combines mean reversion and oscillations. Equation (4.1.2) holds for $0 \leq \zeta < 1$ (underdamped regime), where the process oscillates before stabilizing. For $\zeta > 1$ (overdamped), it decays monotonically, with $\zeta = 1$ marking the transition. The mean reversion rate is set by γ, and ω determines the oscillation frequency.

5 Application

5.1 Data sources

For both unemployment rates[3] and house price indices[4], we used datasets from CBS. For unemployment rates, we sourced both quarterly and yearly data, while for HPI, we used only monthly data. The inclusion of both quarterly and yearly unemployment data was deliberate. The yearly dataset extends back to the 1970s, capturing a broader historical range of economic conditions, whereas the quarterly data, available only from 2003 onward, offers greater temporal granularity.

5.2 Validating Stationarity

Both the quarterly and yearly frequenced unemployment rates (Unemp) exhibit stationarity after first differencing. We determined this by conducting an Augmented Dickey-Fuller (ADF)[5] test and a Breusch-Pagan test[6] for heteroskedasticity. Both of these tests passed with a rejection threshold of 0.05. Below you can find the derived p-values from the data:

Test	Unemp-Q	Unemp-Y	HPI	
Breusch-Pagan	0.3890	0.3807	0.1756	P-value
ADF	$2e^{-5}$	$4.3e^{-5}$	$9.02e^{-8}$	

Table 1: Stationarity P-values

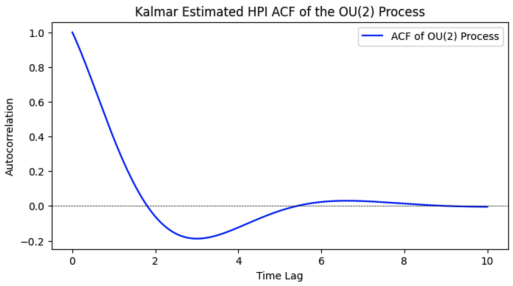
To ensure stationarity, all time series were differenced before testing. For house price indices (HPI), we used second-order differencing to account for long-term trends, while unemployment rates required first-order differencing. The null-hypothesis of the Breush-Pagan is that the differenced indicator is homoscedastic[7]. Conversely, the null-hypothesis of the ADF is that the data is (weakly) stationary[8]. Meaning that after their respective differencing each timeseries is stationary and homoscedastic as desired.

[3] https://opendata.cbs.nl/statline/#/CBS/nl/dataset/80590ned/table?dl=61121
[4] https://opendata.cbs.nl/statline/#/CBS/nl/dataset/84064NED/table?ts=1739636045693
[5] https://en.wikipedia.org/wiki/Augmented_Dickey-Fuller_test
[6] https://en.wikipedia.org/wiki/Breusch-Pagan_test
[7] https://en.wikipedia.org/wiki/Homoscedasticity_and_heteroscedasticity
[8] https://en.wikipedia.org/wiki/Stationary_process

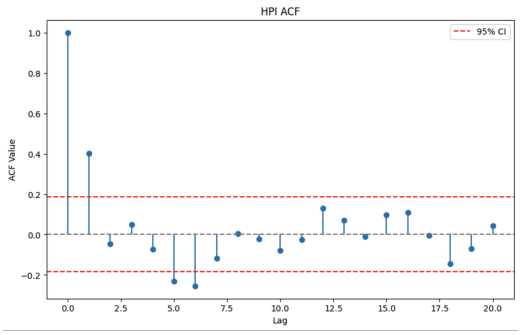
5.3 Empirical vs Theoretical Auto Correlative Structure

Second-Differenced HPI

Appendix Sections §7.2.3 and §7.2.4 provide the implementation of the Kalman filter and its log-likelihood function, while Section §4.1 defines the autocorrelation function (ACF) of the second-order Ornstein-Uhlenbeck (OU) process. As shown below, the Kalman-filtered ACF closely approximates the empirical ACF for the second-differenced HPI:



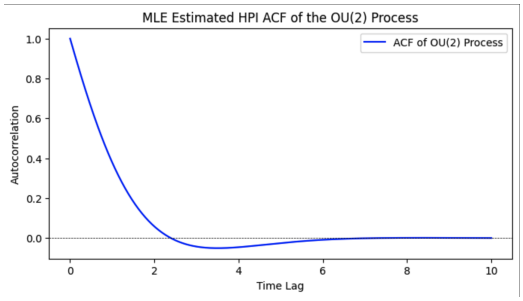
(a) Kalman-Filtered HPI ACF



(b) Empirical HPI ACF

Figure 1

Similarly, Sections §7.2.1 and §7.2.2 provide the MLE-estimated parameters and the Euler-Maruyama approximation of the second-order OU process (Section §3.4). The red line in Figure 1 and subsequent figures indicates the significance threshold for autocorrelation. Compared to the Kalman filter, the MLE approach aligns less precisely with the empirical ACF:

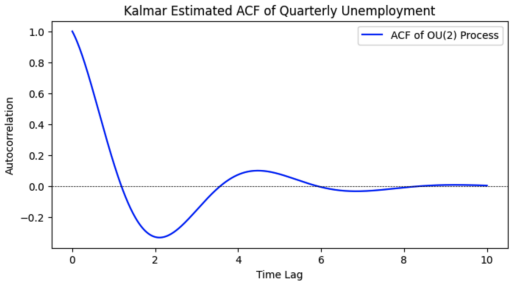


(a) MLE Estimated HPI ACF

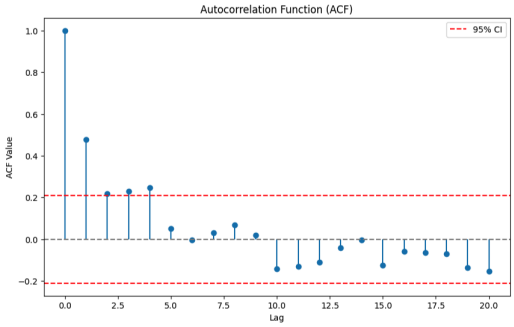
Figure 2

This discrepancy may result from the Kalman filter overfitting the empirical ACF. Whether this is the case will be evaluated using BIC and AIC scores in Section §5.4.

Second-Differenced Quarterly Unemployment



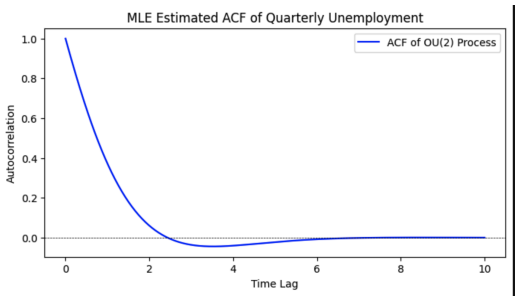
(a) Kalman-Filtered Q-Unemployment ACF



(b) Empirical Q-Unemployment ACF

Figure 3

The ACF reveals distinct differences between the Kalman-filtered and empirical ACFs. Notably, the Kalman-filtered ACF exhibits oscillatory behavior, which is absent in the empirical data. This suggests that the Kalman-based estimation method imposes a structure not inherently present in the observed data.

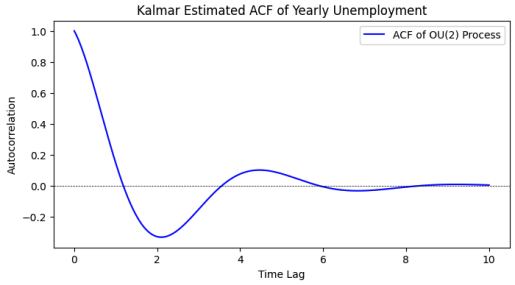


(a) MLE Estimated Q-Unemployment ACF

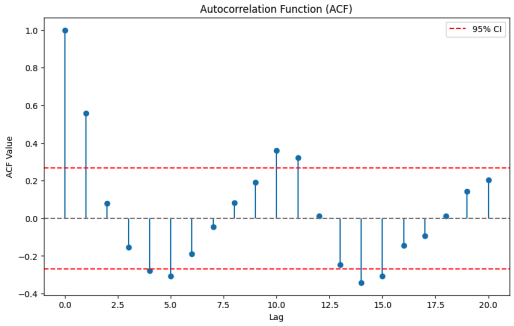
Figure 4

In contrast, the MLE-estimated ACF is more consistent with the empirical ACF, suggesting that the MLE method better captures the underlying temporal dependencies.

First-Differenced Yearly Unemployment



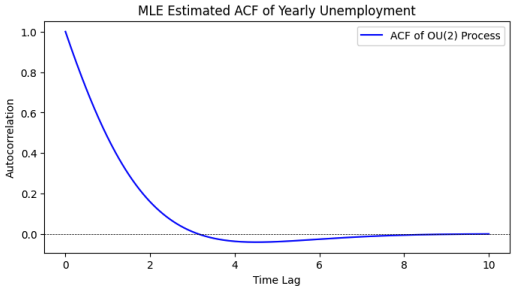
(a) Kalman-Filtered Y-Unemployment ACF



(b) Empirical Y-Unemployment ACF

Figure 5

Here, the ACF shows minimal differences between the Kalman-filtered and empirical ACFs. However, the MLE-estimated ACF exhibits distinct characteristics:



(a) MLE Estimated Y-Unemployment ACF

Figure 6

Unlike the empirical ACF, the MLE-estimated ACF displays notable oscillatory behavior, which is absent in the empirical ACF.

5.4 Model Fit Evaluation

We assess model fit using the Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC)[9], where lower values indicate better fit. Table 2 compares different estimation methodologies.

Method	Kalman	MLE	AR(2)	Data
AIC	−45.40	−683.42	−671.66	HPI
BIC	−37.25	−675.26	−660.79	HPI
AIC	−33.94	−780.97	28.17	Q-Unemployment
BIC	−26.54	−773.57	35.60	Q-Unemployment
AIC	−18.12	−379.37	−398.76	Y-Unemployment
BIC	−12.21	−373.46	−390.80	Y-Unemployment

Table 2: AIC and BIC values for different estimation methods across datasets.

MLE consistently provides the best fit across all datasets, with the lowest AIC and BIC values. For HPI, it outperforms both Kalman and AR(2), which perform moderately well.

For quarterly unemployment, MLE remains superior, while AR(2) performs poorly, suggesting it is unsuitable for short-term dynamics. The Kalman filter, though better than AR(2), still lags behind MLE.

For yearly unemployment, AR(2) slightly outperforms MLE, suggesting simpler models may suffice in some cases. However, the Kalman filter performs the worst in this category, with significantly higher BIC values.

In Summary
MLE is the most robust method overall, yielding the lowest AIC and BIC in most cases. AR(2) performs well for yearly unemployment but fails for quarterly data. The Kalman filter remains viable but underperforms compared to MLE and AR(2), particularly for unemployment modeling. The large discrepancy between the MLE and Kalman filter log-likelihood values suggests that the Kalman-based likelihood optimization may not have fully converged. Further investigation is required to determine whether initialization choices or measurement noise assumptions contribute to this effect.

[9] `https://vitalflux.com/aic-vs-bic-for-regression-models-formula-examples/`

6 Conclusions

In this paper, we developed a framework for macro-economic scenario generation based on Ornstein-Uhlenbeck processes. We began by revisiting the classical first-order OU process and highlighting its relationship to the discrete-time AR(1) model. Building on this foundation, we introduced the second-order OU (OU(2)) process and demonstrated its correspondence to a continuous-time analog of an AR(2) model. We derived analytical expressions for the OU(2) model's behavior (such as its autocorrelation function and stationary variance), illustrating that this extended process is well-suited for capturing the mean-reverting and cyclical patterns observed in economic time series. Finally, through an empirical application, we calibrated the OU(2) model to real macro-economic data and compared its performance against a standard AR(2) model and a state-space Kalman filter approach.

The results of our empirical analysis indicate that the OU(2) model offers strong performance in modeling macro-economic variables. In our case studies, maximum likelihood estimates of the OU(2) provided the best fit to the data in terms of AIC and BIC in most scenarios, effectively capturing both short-term fluctuations and long-term reversion tendencies. Notably, for high-frequency data (such as quarterly unemployment rates), the continuous-time OU(2) approach captured dynamics that a discrete AR(2) model struggled to replicate. In contrast, for lower-frequency data (such as yearly aggregates), the simpler AR(2) remained competitive. The Kalman filter-based estimation of the OU process was viable but tended to underperform compared to direct MLE estimation of OU parameters and even the purely discrete AR(2) model in certain cases. These findings underscore the flexibility and robustness of the OU(2) process in replicating the statistical properties of macro-economic time series, making it a valuable tool for scenario generation.

Looking ahead, the Ornstein-Uhlenbeck framework can be extended in several directions to enhance macro-economic modeling. One potential avenue is to incorporate jumps or non-Gaussian noise (e.g. using Lévy process driven OU variants) to capture sudden economic shocks or heavy-tailed behavior, building on the generalizations discussed by Maller et al. [3]. Another extension could involve coupling multiple OU processes to model multivariate scenarios in which key economic indicators evolve jointly. An example of this idea is to couple unemployment rates with gdp rates using of Okun's Law [5]. Finally, investigating regime-switching OU models where the mean-reversion rate or volatility shifts under different economic conditions could provide further realism in long term simulations. By exploring these enhancements, future research can continue to improve the realism and accuracy of scenario generation for macro-economic risk analysis and planning.

References

[1] E. Kloeden, P.E. Platen. *Numerical Solution of Stochastic Differential Equations.* Springer, 305–331, 1991. ISBN: 3-540-54062-8.

[2] Müller G. Szimayer A. Maller, R.A. *Ornstein-Uhlenbeck processes and extentions.* Handbook of Financial Time Series Analysis 27, 599-612, 2006.

[3] Ross A. Maller. *Ornstein-Uhlenbeck Processes and Extensions.* 2006. Link: https://mediatum.ub.tum.de/doc/1072652/1072652.pdf.

[4] G. N. Milstein. Approximate integration of stochastic differential equations. *Theory of Probability and its Applications*, 19(3):557–562, 1975.

[5] Christopher Neely. Okun's law: Output and unemployment. *Economic Synopses*, 4, 01 2010.

[6] L.S. Ornstein and G.E. Uhlenbeck. On the theory of the brownian motion. *Physical Review*, 36(5):823–841, 1930.

[7] O.A. Vasicek. *An equilibrium characterisation of the term structure.* J. Fin. Econ. 5, 177-188, 1977.

7 Appendix

7.1 Power Spectral Density of the Second-Order OU Process

The second-order Ornstein-Uhlenbeck (OU) process is defined by the following stochastic differential equation:

$$\frac{d^2X_t}{dt^2} + 2\lambda\frac{dX_t}{dt} + \omega_0^2 X_t = \sigma W_t$$

where $\lambda > 0$ is the damping coefficient, $\omega_0 > 0$ is the natural frequency, and W_t is a white noise process with spectral density $S_{\tilde{\xi}(\omega)} = \sigma^2$.

7.1.1 Derivation of the Power Spectral Density

Taking the Fourier transform of both sides of the differential equation:

$$(-\omega^2 + 2i\lambda\omega + \omega_0^2)\tilde{X}(\omega) = \tilde{\xi}(\omega)$$

Solving for $\tilde{X}(\omega)$

$$\tilde{X}(\omega) = \frac{\tilde{\xi}(\omega)}{\omega_0^2 - \omega^2 + 2i\lambda\omega}$$

The power spectral density (PSD) is defined as:

$$S_{\tilde{X}}(\omega) = |H(\omega)|^2 S_X(\omega)$$

where $H(\omega)$ is the transfer function:

$$H(\omega) = \frac{1}{\omega_0^2 - \omega^2 + 2i\lambda\omega}$$

Since W_t is white noise with $S_{\tilde{\xi}(\omega)} = \sigma^2$, we obtain:

$$S_{\tilde{X}}(\omega) = \sigma^2 \left| \frac{1}{\omega_0^2 - \omega^2 + 2i\lambda\omega} \right|^2$$

Computing the magnitude squared:

$$|H(\omega)|^2 = \frac{1}{(\omega_0^2 - \omega^2)^2 + 4\lambda^2\omega^2}$$

we find the PSD:

$$S_{\tilde{X}}(\omega) = \frac{\sigma^2}{(\omega_0^2 - \omega^2)^2 + 4\lambda^2\omega^2} \tag{7.1.1}$$

7.1.2 Derivation of the Autocorrelation Function

By definition, the autocorrelation function $R_X(\tau)$ is given by the inverse Fourier transform[10]:

$$R_X(\tau) = \frac{1}{2\pi} \int_{-\infty}^{\infty} S_{\tilde{X}}(\omega) e^{i\omega\tau} d\omega$$

Substituting $S_{\tilde{X}}(\omega)$,

$$R_X(\tau) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \frac{\sigma^2}{(\omega_0^2 - \omega^2)^2 + 4\lambda^2\omega^2} e^{i\omega\tau} d\omega$$

The integral can be evaluated using contour integration[11] by noting the complex poles at:

$$\omega = \pm(\omega_d + i\lambda), \quad \omega = \pm(\omega_d - i\lambda)$$

where $\omega_d = \sqrt{\omega_0^2 - \lambda^2}$ is the damped frequency. Applying the residue theorem[12], the result is:

$$R_X(\tau) = Ae^{-\lambda|\tau|} \cos(\omega_d|\tau|) \tag{7.2.1}$$

where A is a proportionality constant. This function describes an exponentially damped oscillation, characterizing the memory decay of the process.

[10] https://mathworld.wolfram.com/FourierTransformInverseFunction.html
[11] https://mathworld.wolfram.com/ContourIntegration.html
[12] https://mathworld.wolfram.com/ResidueTheorem.html

7.2 Code Snippets

7.2.1 MLE OU Parameter Estimation

```python
def estimate_ou_parameters(data, dt):
    """
    Estimates the parameters of the second-order OU process using maximum likelihood.

    Parameters:
    data : array-like
        Observed time series data.
    dt : float
        Time step size.

    Returns:
    dict
        Estimated parameters { "gamma": gamma, "omega": omega, "sigma": sigma }.
    """

    # Initial guess based on reasonable assumptions
    initial_guess = [1.5, -0.5, 0.1]  # Approximate AR(2) coefficients and sigma

    # Optimization bounds to enforce stability conditions
    bounds = [(None, None), (None, None), (1e-5, None)]  # sigma must be positive

    # Minimize negative log-likelihood
    result = minimize(second_order_ou_likelihood, initial_guess, args=(data, dt),
                      bounds=bounds, method='L-BFGS-B')

    if result.success:
        a1, a2, sigma = result.x
        gamma = (2 - a1) / dt
        omega = np.sqrt(-(a1 + a2 - 2) / dt**2)

        return {"gamma": gamma, "omega": omega, "sigma": sigma}
    else:
        raise ValueError("Optimization failed. Try different initial conditions.")
```

7.2.2 Second Order OU Euler-Maruyama Log-likelihood

```python
def second_order_ou_likelihood(params, data, dt):
    """
    Computes the negative log-likelihood of the second-order OU process
    (discretized to an AR(2) structure) given parameters and time-series data.

    Parameters:
    params : list
        [a1, a2, sigma] where:
        - a1 = 2 - gamma * dt - omega^2 * dt^2
        - a2 = - (1 - gamma * dt)
        - sigma is the process noise intensity.
    data : array-like
        Observed time series data.
    dt : float
        Time step size.

    Returns:
    float
        Negative log-likelihood value.
    """
    a1, a2, sigma = params
    n = len(data)

    # Ensure parameters satisfy model constraints (positivity)
    if sigma <= 0:
        return np.inf

    # Initialize likelihood computation
    residuals = np.zeros(n - 2)
    for t in range(2, n):
        predicted_Xt = a1 * data[t - 1] + a2 * data[t - 2]
        residuals[t - 2] = data[t] - predicted_Xt

    # Gaussian log-likelihood computation
    variance = sigma**2 * dt
    log_likelihood = -0.5 * np.sum(np.log(2 * np.pi * variance) + (residuals**2 / variance))

    return -log_likelihood  # Negative because we minimize
```

7.2.3 Second Order OU Kalman Filter Log-likelihood

```python
def Kalman_filter_likelihood(params, data, dt):
    """
    Compute the negative log-likelihood using a Kalman filter for a second-order OU process.

    Parameters:
        params (list): [a1, a2, sigma], the parameters to estimate.
        data (np.ndarray): Time series data (observed values).
        dt (float): Time step between observations.

    Returns:
        float: Negative log-likelihood.
    """
    a1, a2, sigma = params

    # Ensure stability of the system
    if a1 <= 0 or a2 <= 0 or sigma <= 0:
        return np.inf

    # Initinialize Parameter Estimations
    n = len(data)
    A = np.array([[1, dt], [-a1 * dt, 1 - a2 * dt]])
    B = np.array([[0], [sigma * np.sqrt(dt)]])
    C = np.array([[1, 0]])
    X_t = np.array([[data[0]], [(data[1] - data[0]) / dt]])
    P_t = np.eye(2) * 1e-3
    Q, R = B @ B.T, 1e-3
    log_likelihood = 0

    for t in range(1, n):
        # Prediction step
        X_pred = A @ X_t
        P_pred = A @ P_t @ A.T + Q

        # Compute Kalman gain
        S = C @ P_pred @ C.T + R
        K = P_pred @ C.T @ np.linalg.inv(S)

        # Update step (Fix applied here)
        residual = data[t] - (C @ X_pred)[0, 0]
        X_t = X_pred + K * residual  # Element-wise multiplication instead of matrix multiplication
        P_t = (np.eye(2) - K @ C) @ P_pred

        # Log-likelihood update
        log_likelihood += 0.5 * (np.log(2 * np.pi * S) + (residual**2) / S)

    return log_likelihood  # Negative log-likelihood for minimization
```

7.2.4 Second Order OU Kalman Estimation

```
    def estimate_ou_Kalman_parameters(df, value_col, time_col='pit_date'):
    """
    Estimate second-order OU parameters using Kalman filtering.

    Parameters:
        df (pd.DataFrame): DataFrame with a time column and observed values
    .
        value_col (str): Column name for the observed time series values.
        time_col (str): Column name for timestamps (default: 'pit_date').

    Returns:
        dict: Estimated parameters {'a1', 'a2', 'sigma'}.
    """
    # Sort DataFrame by time
    df = df.sort_values(by=time_col)

    # Extract time steps (assuming uniform spacing)
    # time_values = df[time_col].values
    dt = 1  # Compute average time step

    # Extract time series data
    data = df[value_col].values

    # Initial guesses for parameters [a1, a2, sigma]
    initial_guess = [5, 0.5, 0.005]

    # Minimize negative log-likelihood using Kalman filtering
    result = minimize(
        Kalman_filter_likelihood,
        x0=initial_guess,
        args=(data, dt),
        bounds=[(1e-5, None), (1e-5, None), (1e-5, None)],  # Ensure
    parameters remain positive
        method='L-BFGS-B',
    )

    # Extract results
    if result.success:
        a1, a2, sigma = result.x
        return {'a1': a1, 'a2': a2, 'sigma': sigma}
    else:
        raise ValueError("Parameter estimation failed: " + result.message)
```