

Семинар 8

```
[1]: import matplotlib.pyplot as plt
import numpy as np
from quantecon import LinearStateSpace

plt.rcParams["figure.figsize"] = (11, 5) # set default_
    ↪ figure size
```

1. Линейная модель пространства состояний

Данный класс моделей можно рассматривать как обобщение идеи авторегрессионных процессов. Он находит свое применение в описании большого числа моделей и решении большого числа задач:

- Модель постоянного дохода Фридмана.
- Модель сглаживания сборов налогов Барро.
- Версия модели гиперинфляции Кагана для рациональных ожиданий.
- И так далее.

Не откладывая в долгий ящик, запишем общий вид системы:

$$\begin{aligned}x_{t+1} &= Ax_t + C\omega_{t+1} \\ y_t &= Gx_t + Hv_t \\ x_0 &\sim N(\mu_0, \Sigma_0) \\ \omega_t &\sim N(0, I_m) \\ v_t &\sim N(0, I_\ell)\end{aligned}$$

Подробнее о компонентах модели:

- x_t — вектор размера $n \times 1$. Это вектор **состояния** системы.
- y_t — вектор размера $k \times 1$. Это вектор **измерений**.
- $A_{n \times n}$ — **матрица перехода** от текущего состоянию к следующему. Также может называться **матрицей системы**.
- $C_{n \times m}$ — **матрица волатильности**, также может называться **входной матрицей**.
- $G_{k \times n}$ — **матрица выхода** или **матрица измерений**.
- $H_{k \times \ell}$ — **матрица шума**. Если $\omega_t \equiv v_t$, то матрица называется **матрицей прямой связи**.

Шоки ω_t не обязаны быть IID. Достаточно, чтобы они были последовательностью разности мартингалов, то есть чтобы выполнялось

$$E(\omega_{t+1} \mid x_t, x_{t-1}, \dots) = 0$$

1.1. Распределения и моменты

1.1.1. Моменты

Выражения для безусловных средних найти легко. Обозначим $\mu_t := E(x_t)$. Тогда, в силу линейности, получим

$$\mu_{t+1} = A\mu_t \quad (1)$$

Ковариационная матрица x_t определяется как

$$\Sigma_t := E[(x_t - \mu_t)(x_t - \mu_t)']$$

Отсюда получим

$$x_{t+1} - \mu_{t+1} = A(x_t - \mu_t) + C\omega_{t+1}$$

$$\Sigma_{t+1} = A\Sigma_t A' + CC' \quad (2)$$

Автоковариационная матрица определяется как

$$\Sigma_{t+j,t} := E[(x_{t+j} - \mu_{t+j})(x_t - \mu_t)']$$

что, после несложных расчетов, приводит к

$$\Sigma_{t+j,t} = A^j \Sigma_t$$

Аналогичные значения для y_t равны

$$E(y_t) = E(Gx_t + Hv_t) = G\mu_t$$

$$\text{Var}(y_t) = \text{Var}(Gx_t + Hv_t) = G\Sigma_t G' + HH'$$

1.1.2. Распределения

Если инновации в модели имеют нормальное распределение, то

$$\begin{aligned} x_t &\sim N(\mu_t, \Sigma_t) \\ y_t &\sim N(G\mu_t, G\Sigma_t G' + HH') \end{aligned}$$

Часто бывает нужно найти совместное распределение последовательности состояний. Плотность, в силу Марковости, определяется по формуле

$$\begin{aligned} p(x_0, x_1, \dots, x_T) &= p(x_0) \prod_{t=0}^{T-1} p(x_{t+1} | x_t) \\ p(x_0) &\leftarrow N(\mu_0, \Sigma_0) \\ p(x_{t+1} | x_t) &\leftarrow N(A\mu_t, A\Sigma_t A' + CC') \end{aligned}$$

Рассмотрим нахождение описанных выше распределений примере процесса AR(4). Построим функцию, выводящую графики симуляций этого процесса.

```

[2]: def cross_section_plot(
    A,
    C,
    G,
    T=20,
    ymin=-0.8,
    ymax=1.25,
    simulations=20,
    n=4,
):
    ar = LinearStateSpace(A, C, G, mu_0=np.ones(n))

    fig, axes = plt.subplots(1, 2, figsize=(16, 5))

    for ax in axes:
        ax.grid(alpha=0.4)
        ax.set_ylim(ymin, ymax)

    ax = axes[0]
    ax.set_ylim(ymin, ymax)
    ax.set_ylabel("$y_t$", fontsize=12)
    ax.set_xlabel("time", fontsize=12)
    ax.vlines((T,), -1.5, 1.5)

    ax.set_xticks((T,))
    ax.set_xticklabels(("T",))

    sample = []
    for i in range(simulations):
        rcolor = np.random.choice(("c", "g", "b", "k"))
        x, y = ar.simulate(ts_length=T + 15)
        y = y.flatten()
        ax.plot(y, color=rcolor, lw=1, alpha=0.5)
        ax.plot((T,), (y[T],), "ko", alpha=0.5)
        sample.append(y[T])

    y = y.flatten()
    axes[1].set_ylim(ymin, ymax)
    axes[1].set_ylabel("$y_t$", fontsize=12)
    axes[1].set_xlabel("relative frequency", fontsize=12)
    axes[1].hist(sample, bins=16, density=True,
orientation="horizontal", alpha=0.5)
    plt.show()

```

Зададим параметры модели.

```

[3]: phi_1, phi_2, phi_3, phi_4 = 0.5, -0.2, 0, 0.5
    sigma = 0.1

    A = [

```

```

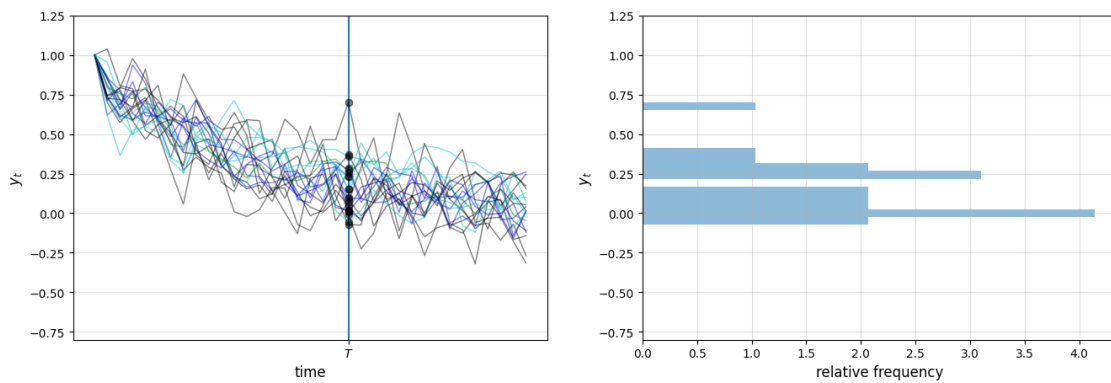
[phi_1, phi_2, phi_3, phi_4],
[1, 0, 0, 0],
[0, 1, 0, 0],
[0, 0, 1, 0],
]

C = [
[sigma],
[0],
[0],
[0],
]

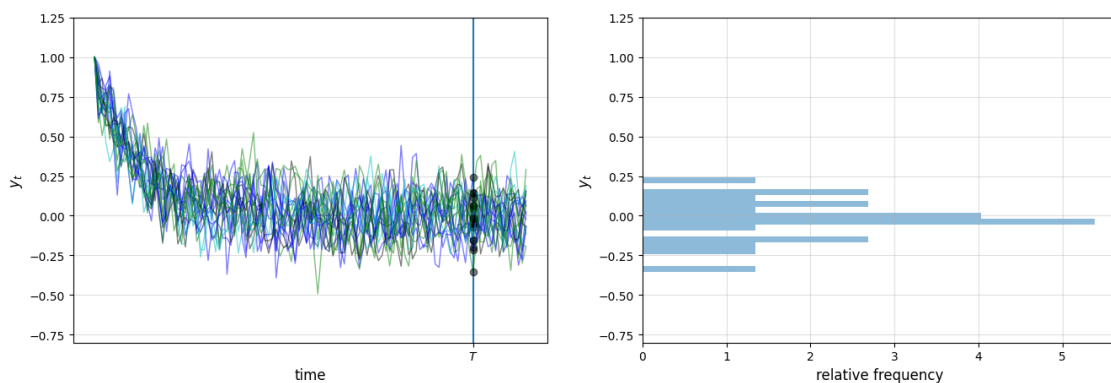
G = [1, 0, 0, 0]

```

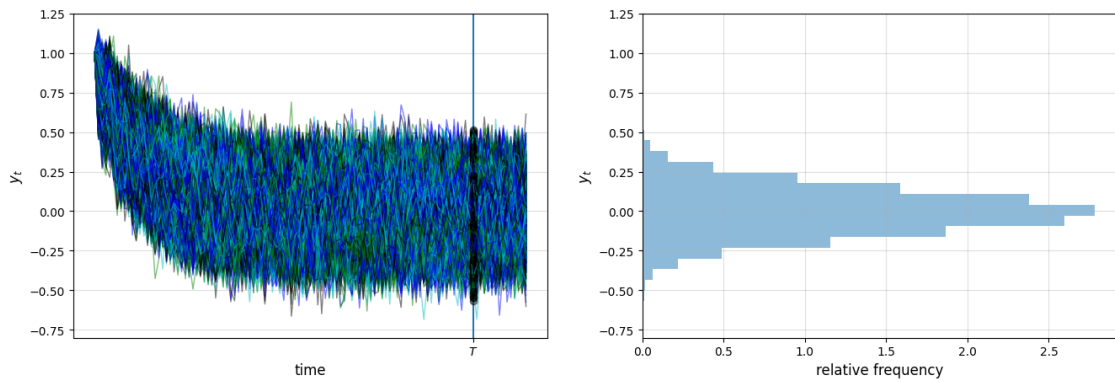
[4]: `cross_section_plot(A, C, G)`



[5]: `t = 100`
`cross_section_plot(A, C, G, T=t)`



[6]: `t = 100`
`cross_section_plot(A, C, G, simulations=10_000, T=t)`



1.2. Стационарные распределения

Распределение ψ_∞ назовем стационарным, если

$$x_t \sim \psi_\infty, \quad x_{t+1} = Ax_t + C\omega_{t+1} \Rightarrow x_{t+1} \sim \psi_\infty$$

В случае нормальности инноваций имеем

$$\psi_\infty = N(\mu_\infty, \Sigma_\infty)$$

где μ_∞ и Σ_∞ — неподвижные точки (1) и (2), соответственно.

1.2.1. Условие стационарности

Уравнение (1) будет иметь единственную неподвижную точку, если **все** собственные значения матрицы A по абсолютному значению строго меньше 1. В этом случае уравнение (2) также будет иметь единственную неподвижную точку.

Более того, вне зависимости от начального распределения

$$\begin{aligned} \mu_t &\rightarrow \mu_\infty = 0 \\ \Sigma_t &\rightarrow \Sigma_\infty \\ t &\rightarrow \infty \end{aligned}$$

1.2.2. Система с постоянной компонентой

Если вектор состояния имеет постоянную компоненту, например

$$x_t = \begin{pmatrix} 1 \\ y_t \\ y_{t-1} \end{pmatrix}$$

то $\mu_t \rightarrow 0$!

Реорганизуем систему так, чтобы компоненты можно было представить в виде

$$A = \begin{pmatrix} A_1 & a \\ 0 & 1 \end{pmatrix}, \quad C = \begin{pmatrix} C_1 \\ 0 \end{pmatrix}, \quad x_t = \begin{pmatrix} x_{1t} \\ 1 \end{pmatrix}$$

где x_{1t} и a — столбцы длины $n - 1$. Получим

$$x_{1,t+1} = A_1 x_{1t} + a + C_1 \omega_{t+1}$$

$$\mu_{1,t+1} = A_1 \mu_{1t} + a$$

$$\mu_{1\infty} = (I - A_1)^{-1} a$$

$$\mu_{\infty} = \begin{pmatrix} \mu_{1\infty} \\ 1 \end{pmatrix}$$

1.2.3. Эргодичность

Свойство эргодичности означает, что выборочные средние сходятся к своим математическим ожиданиям в стационарном распределении:

- $\frac{1}{T} \sum_{t=1}^T x_t \rightarrow \mu_{\infty}$
- $\frac{1}{T} \sum_{t=1}^T (x_t - \bar{x}_T)(x_t - \bar{x}_T)' \rightarrow \Sigma_{\infty}$
- $\frac{1}{T} \sum_{t=1}^T (x_{t+j} - \bar{x}_T)(x_t - \bar{x}_T)' \rightarrow A^j \Sigma_{\infty}$

В случае нормальных инноваций любой стационарный процесс — эргодичен.

1.3. Построение прогнозов

Прогноз x_t на один шаг вперед равен

$$E_t(x_{t+1}) = E(x_{t+1} | x_t, x_{t-1}, \dots) = E(x_{t+1} | x_t) = Ax_t$$

Ошибка прогноза и ее ковариационная матрица равны

$$\begin{aligned} \Xi &= x_{t+1} - E_t(x_{t+1}) = C\omega_{t+1} \\ V_1 &= CC' \end{aligned}$$

Прогноз на j шагов вперед равен

$$E_t(x_{t+j}) = E_t(A^j x_t + A^{j-1} C \omega_{t+1} + A^{j-2} C \omega_{t+2} + \dots + A^0 C \omega_{t+j}) = A^j x_t$$

Ковариационная матрица ошибки прогноза равна

$$\begin{aligned} V_j &= \sum_{k=0}^{j-1} A^k CC' (A^k)' \\ V_j &= CC' + AV_{j-1}A' \end{aligned}$$

Если **все** собственные значения матрицы A по абсолютному значению строго меньше 1, то V_j будет сходиться к

$$V_{\infty} = CC' + AV_{\infty}A'$$

2. Эксперименты

2.1. Разностное уравнение второго порядка

Пусть $\{y_t\}$ — детерминированное разностное уравнение второго порядка

$$y_{t+1} = \phi_0 + \phi_1 y_t + \phi_2 y_{t-1}$$

В LSS виде это уравнение может быть записано при помощи следующих обозначений

$$x_t = \begin{pmatrix} 1 \\ y_t \\ y_{t-1} \end{pmatrix}, \quad A = \begin{pmatrix} 1 & 0 & 0 \\ \phi_0 & \phi_1 & \phi_2 \\ 0 & 1 & 0 \end{pmatrix}, \quad C = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \quad G = \begin{pmatrix} 0 & 1 & 0 \end{pmatrix}$$

Убедимся в справедливости данного представления. Очевидно, что первое уравнение системы примет вид

$$x_{t+1} = Ax_t$$

Найдем значение Ax_t

$$Ax_t = \begin{pmatrix} 1 & 0 & 0 \\ \phi_0 & \phi_1 & \phi_2 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ y_t \\ y_{t-1} \end{pmatrix} = \begin{pmatrix} 1 \\ \phi_0 + \phi_1 y_t + \phi_2 y_{t-1} \\ y_t \end{pmatrix} = \begin{pmatrix} 1 \\ y_{t+1} \\ y_t \end{pmatrix} = x_{t+1}$$

Значение наблюдения при этом равно

$$Gx_{t+1} = \begin{pmatrix} 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ y_{t+1} \\ y_t \end{pmatrix} = y_{t+1} = \phi_0 + \phi_1 y_t + \phi_2 y_{t-1}$$

```
[7]: def plot_lss(A, C, G, ts_length=50, mu_0=None):
    n = len(A)

    if mu_0 is None:
        mu_0 = np.ones(n)

    ar = LinearStateSpace(A, C, G, mu_0=mu_0)
    x, y = ar.simulate(ts_length)
    y = y.flatten()

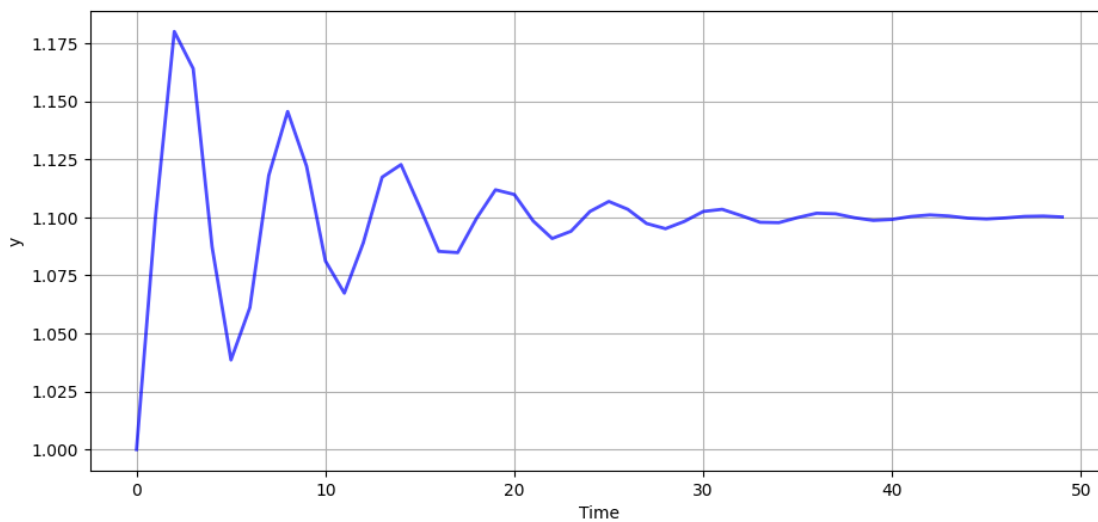
    fig, (ax) = plt.subplots()
    ax.grid()
    ax.plot(y, "b-", lw=2, alpha=0.7)
    ax.set_xlabel("Time")
    ax.set_ylabel("y")
    plt.show()
```

```
[8]: phi_0, phi_1, phi_2 = 1.1, 0.8, -0.8

A = [
    [1, 0, 0],
    [phi_0, phi_1, phi_2],
    [0, 1, 0],
]

C = np.zeros((3, 1))
G = [0, 1, 0]

plot_lss(A, C, G)
```



2.2. Процесс авторегрессии

Рассмотрим процесс AR(4)

$$y_{t+1} = \phi_1 y_t + \phi_2 y_{t-1} + \phi_3 y_{t-2} + \phi_4 y_{t-3} + \sigma \omega_{t+1}$$

$$\omega_t \sim \text{iid } N(0, I)$$

Представим модель в виде LSS

$$x_t = \begin{pmatrix} y_t \\ y_{t-1} \\ y_{t-2} \\ y_{t-3} \end{pmatrix}, \quad A = \begin{pmatrix} \phi_1 & \phi_2 & \phi_3 & \phi_4 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \quad C = \begin{pmatrix} \sigma \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad G = \begin{pmatrix} 1 & 0 & 0 & 0 \end{pmatrix}$$

Матрица A называется **матрицей Фробениуса** или **сопровождающей матрицей полинома**

$$y_{t+1} - \phi_1 y_t - \phi_2 y_{t-1} - \phi_3 y_{t-2} - \phi_4 y_{t-3}$$

```
[9]: phi_1, phi_2, phi_3, phi_4 = 0.5, -0.2, 0, 0.5
      sigma = 0.2
```

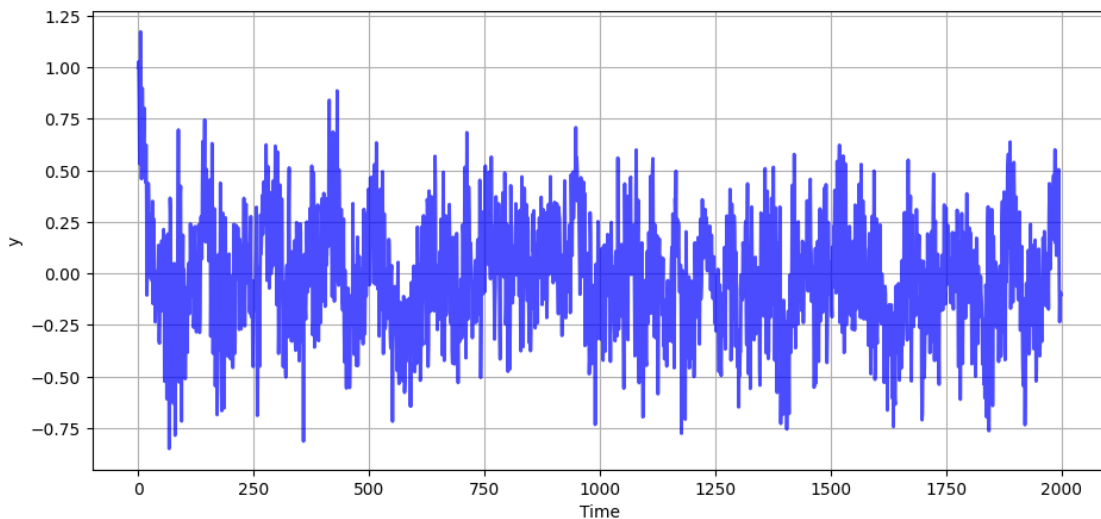
```
A_1 = [
    [phi_1, phi_2, phi_3, phi_4],
    [1, 0, 0, 0],
    [0, 1, 0, 0],
    [0, 0, 1, 0],
]
```

```
C_1 = [
    [sigma],
    [0],
    [0],
    [0],
]
```



```
]
G_1 = [1, 0, 0, 0]

plot_1ss(A_1, C_1, G_1, ts_length=2000)
```



2.3. Сезонность

Мы можем определить два типа сезонности:

- детерминированная $y_t = y_{t-4}$
- недетерминированная $y_t = \phi_4 y_{t-4} + \sigma \omega_t$

Матрица A имеет вид

$$A = \begin{pmatrix} 0 & 0 & 0 & \phi_4 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \quad C = \begin{pmatrix} \sigma \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad G = (1 \ 0 \ 0 \ 0)$$

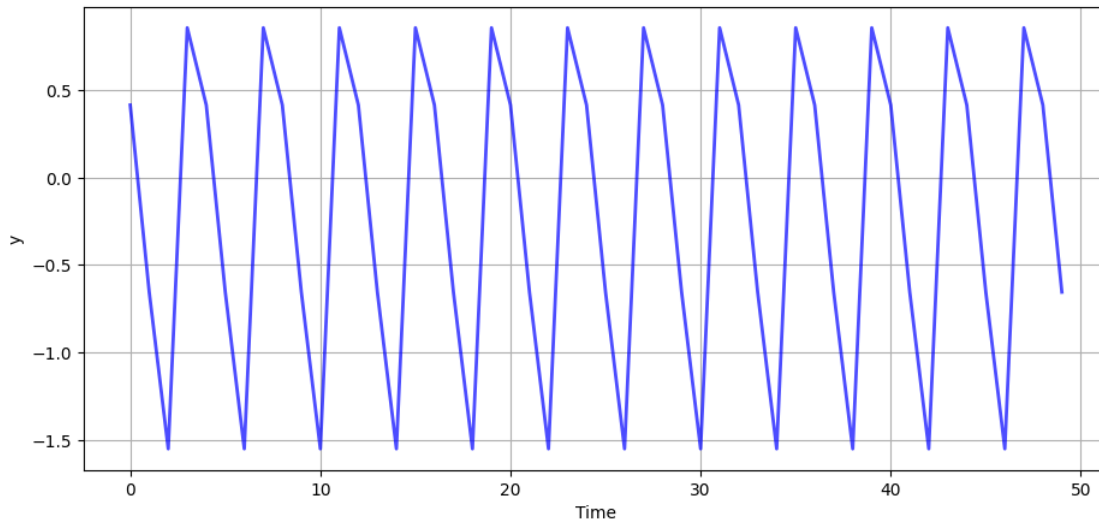
причем для детерминированного случая $\phi_4 = 1, \sigma = 0$.

```
[10]: A_ds = (
        (0, 0, 0, 1),
        (1, 0, 0, 0),
        (0, 1, 0, 0),
        (0, 0, 1, 0),
    )
C_ds = np.zeros((4, 1))

A_us = (
        (0, 0, 0, 0.8),
        (1, 0, 0, 0),
        (0, 1, 0, 0),
        (0, 0, 1, 0),
    )
```

```
)  
C_us = [[0.1], [0], [0], [0]]  
  
G = [1, 0, 0, 0]
```

```
[11]: plot_lss(A_ds, C_ds, G, mu_0=np.random.normal(size=4))
```



```
[12]: plot_lss(A_us, C_us, G, ts_length=150)
```

