# Chapter — Foundations of Phasor Rotation and Projection in Time & Frequency

*How advancing in time and angle on the unit circle reveals the cosine and sine components of a signal*

---

## 1. The Core Idea

When we multiply a signal $x[n]$ by a **complex exponential**:

$$e^{j2\pi f_m n/f_s}$$

we are **rotating a phasor** around the **unit circle** at a speed determined by $f_m$ (mixer frequency) while **sampling in discrete time** $n$ at intervals of $1/f_s$.

---

## 2. Breaking it Down

### a. Time step per sample

- Each sample $n$ occurs at time:

$$t_n = \frac{n}{f_s}$$

- The **gap** between samples = $1/f_s$ seconds.

### b. Rotation step per sample

- The **phase increment** per sample is:

$$\Delta\theta = \frac{2\pi f_m}{f_s} \quad \text{(in radians)}$$

- This tells us **how much the phasor rotates on the unit circle between two samples**.

---

## 3. Visualizing the Journey

Think of it like this:

1. **Time Axis (horizontal)**:
    - You step forward in **equal time steps** of $1/f_s$ seconds.
2. **Unit Circle (phase space)**:
    - At each step, you rotate the arrow by $\Delta\theta$ radians.
    - After several steps, you complete a full revolution if the total phase = $2\pi$ radians.

---

## 4. What's Happening with the Multiplication

When you multiply $x[n]$ by $e^{j2\pi f_m n/f_s}$:

- **Real part (cosine)** → captures **in-phase component** of $x[n]$
- **Imag part (sine)** → captures **quadrature component** of $x[n]$

Mathematically:

$$x[n] \cdot e^{j2\pi f_m n/f_s} = x[n] \cdot [\cos(2\pi f_m n/f_s) + j\sin(2\pi f_m n/f_s)]$$

So you're really **projecting** $x[n]$ onto two perpendicular axes:

- Cos axis → "how much like cosine" the signal is.
- Sin axis → "how much like sine" the signal is.

---

## 5. A Simple Code View

```
import numpy as np

fs = 8            # Sampling frequency
fm = 2            # Mixer frequency
n = np.arange(4) # 4 samples

t = n / fs
phase = 2 * np.pi * fm * t

m = np.exp(1j * phase)

print("Time (s):", t)
print("Phase (rad):", phase)
print("Complex exponential:", m)
```

### Sample Output

```
Time (s): [0.    0.125 0.25  0.375]
Phase (rad): [0.    1.571 3.142  4.712]
Complex exponential: [1.000+0.000j  0.000+1.000j  -1.000+0.000j  -0.000-1.000j]
```

---

## 6. Key Takeaways

- **Time advances** in $1/f_s$ steps.
- **Phase advances** by $2\pi f_m/f_s$ radians per step.
- Real part = **cos projection**, Imag part = **sin projection**.
- The process is the backbone of **mixing, modulation, and demodulation** in DSP.

---