Signals & Systems I

Labs — Final Code Reference

1. Harmonic (Additive) Synthesis

```
def synthesise(f0, phi, Ak, t):
    y = np.zeros_like(t)
    for k in range(1, len(Ak) + 1):
        y += Ak[k-1] * np.cos(
            2*np.pi*k*f0*t + k*phi - (k-1)*np.pi/2
        )
    return y
```

2. Averaging Filter (FIR Low-Pass)

```
def averaging_filter(x, N):
    return np.convolve(x, np.ones(N)/N, mode="same")
```

3. FIR Envelope Extractor

```
def envelope(x, N):
    w = np.abs(x)
    return np.convolve(w, np.ones(N)/N, mode="same")
```

4. FIR Notch (Nulling) Filter

```
def notch(x, fn, fs):
    wn = 2*np.pi*fn/fs
    return np.convolve(
        x, [1, -2*np.cos(wn), 1], mode="same"
    )
```

5. IIR Envelope Extractor

```
def envelope_iir(x, r):
    w = np.abs(x)
    G = (1-r)/2
    b = [G, G]
    a = [1, -r]
    return signal.lfilter(b, a, w)
```

6. IIR Band-Pass (Resonator)

```
def band_pass(x, fn, fs, r):
    wn = 2*np.pi*fn/fs
    b = [1, 0, -1]
    a = [1, -2*r*np.cos(wn), r**2]
    return signal.lfilter(b, a, x)
```