







## Výsledek filtrace

Oproti minulé filtrace vypadá tento signál lépe, nicméně se mi nepodařilo úspěšně odstranit rušivé frekvence, nýbrž jen utlumit. Měl jsem větší úspěchy, když jsem utlumil 5 za sebou jdoucích indexů (pro index rušivé frekvence 43 jsem tlumil - nastavil frekvenční charakteristiku na 0 - na indexech 41 až 45). Nicméně to by měla vyřešit třetí filtrace.

## Návrh 4 pásmových zadržů

Poslední filtr se skládá ze 4 pásmových zadržů. Pro vygenerování zadržů využijeme funkce `scipy.signal.butterd` a `scipy.signal.butter`.

```
In [510] from scipy.signal import butterd, butter

In [511] # seznam koeficientu jednotlivých filtru
b_a = []

# Nyquistova frekvence
nyq = fs / 2

# generování pásmových zadržů
for i in range(len(cos_freq)):
    # propustné pásmo
    wp = np.array([cos_freq[i] - 30, cos_freq[i] + 30])
    # zaverne pásmo
    ws = np.array([wp[0] - 50, wp[1] + 50])
    # výpočet koeficientu
    N, Wn = butterd(wp / nyq, ws / nyq, 3, 40)
    b, a = butter(N, Wn, 'bandstop')

# impulsní odezva filtru
nof_imp = 32
imp = [1, *np.zeros(nof_imp-1)]
h = lfilter(b, a, imp)

# vykreslení grafu
plt.figure(figsize=(10, 5))
plt.stem(np.arange(len(h)), h)
plt.gca().set_title('Impulsní odezva filtru %d' % (i+1))
b_a.append((b, a))
```

