UED SIM - Algorithmique

Annexe pour l'épreuve sur machine du jeudi 24 mai 2012

Diagramme UML des 3 classes DiscreteTime, DigitalSignal et SampledSignal

DiscreteTime - to : float - dt : float - nb: int + DiscreteTime(n : int, step : float=1.0, start : float=0.0) + start(): float + step(): float + nbVal(): int + setPeriod(newStep : float) + timeValues(): list + t(i:int):float + last(): float + delay(delayVal : float) **DigitalSignal** - vX: list - vZ : list + DigitalSignal(deltaT : float=1., startT : float=0., values : list=[0.0]) + sigValues(): list + FTValues(): list + parameters() : tuple + samplingFreq() : float + deltaF(): float + frequencies(): list + setPeriod(newP : float) + delay(delayVal : float)

Commentaires

to est le premier instant
dt est le pas entre deux instants
successifs
nb est le nombre d'instants

vX est la liste des valeurs du signal vZ est la liste des valeurs de sa transformée de Fourier

Attention: les méthodes setPeriod() et delay() sont redéfinies pour la classe-fille (polymorphisme)

SampledSignal

color: str='b', marker: str='o', ... etc.)

color: str='b', marker: str='o', ... etc.)

- s : str

- updateFT()

- + SampledSignal(expression : str, n : int, Ts : float=1.0, start : float=0.0)
- + analogSignal(): str
- + expression(): str
- + parameters(): tuple
- + delay(delayVal : float)
- + setPeriod(newTs : float)
- $\ sample Signal (n:int=None, Ts:float=None, start:float=None) \\$
- + signalPlot(show : bool=True, noFig : int=1, size : float=5.0, color : str='b', marker : str='s', linecolor : str='g'

linewidth : float=1.0, ...etc.)

+ zeroPadding(nbZeros : int=0) : DigitalSignal

+ signalPlot(show : bool=True, noFig : int=1, size : float=5.0,

+ spectrumPlot(show : bool=True, noFig : int=2, size : float=5.0,

s est une chaîne de caractères donnant l'expression formelle du signal analogique, fonction du temps t, en langage Python

Attention: les méthodes setPeriod(), delay(), parameters() et signalPlot() sont redéfinies pour la classe-fille (polymorphisme)