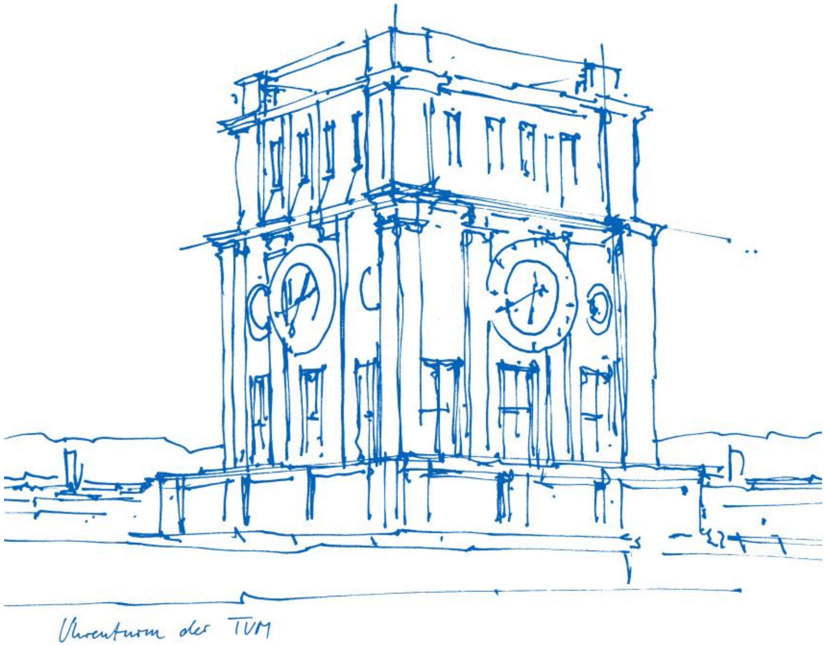
# Fast harmonic estimation using real-time embedded controllers in CoSES smart grid

Erhan Sezgin*, Anurag Mohapatra†, Vedran Peric†

†Center for Combined Smart Energy Systems (CoSES),
Munich School of Engineering

*Gazi University, Ankara



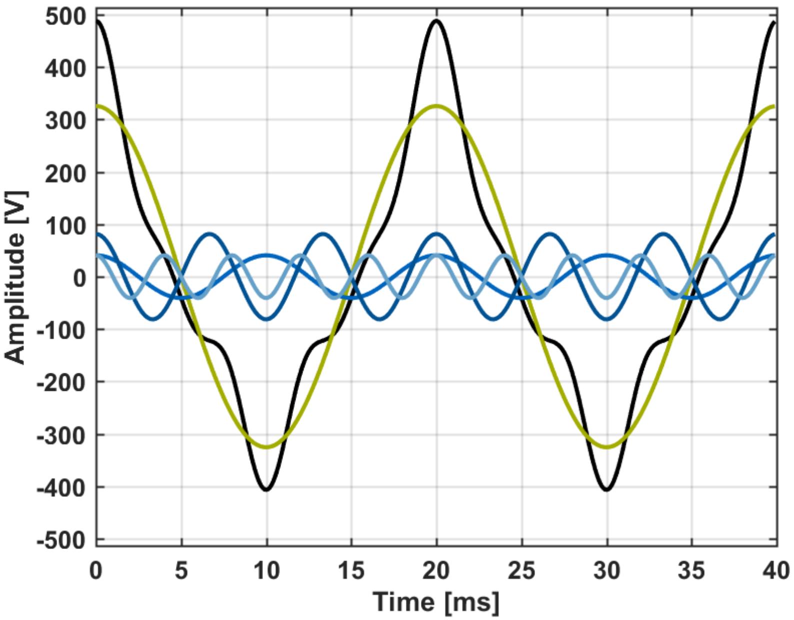
# Outline

- Motivation
- Harmonic analysis methods
- Real time processing on HIL targets
- Conclusions and future work

# Motivation

## What are harmonics:

Integer & non-integer(inter-harmonics) multiples of fundamental component(50Hz) of a voltage or current signal.




Fundamental, harmonics & **overall** signal.

## Caused by:

- Power electronic devices
- Non-linear loads
- Unbalanced grids due to high penetration of renewables.

## Important for:

- Grid monitoring
- Application of legal sanctions defined by regulations
- Active power filtering

# Motivation – within CoSES

*How to provide accurate feedback of harmonic presence in the measurements for microgrid experiments?*

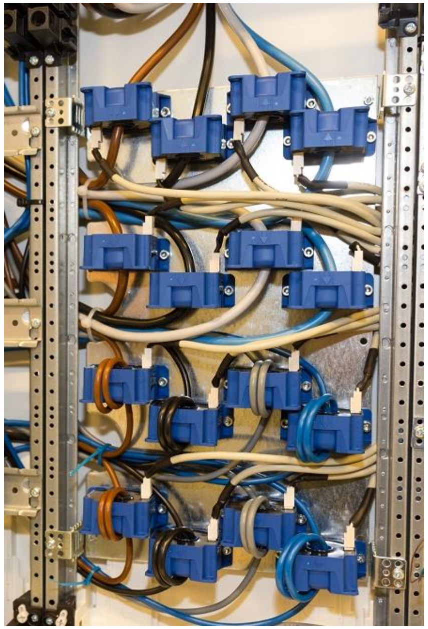
# Motivation – within CoSES

*How to provide accurate feedback of harmonic presence in the measurements for microgrid experiments…while not burdening the real-time controller?*

# Motivation – within CoSES

*How to provide accurate feedback of harmonic presence in the measurements for microgrid experiments…while not burdening the real-time controller?*

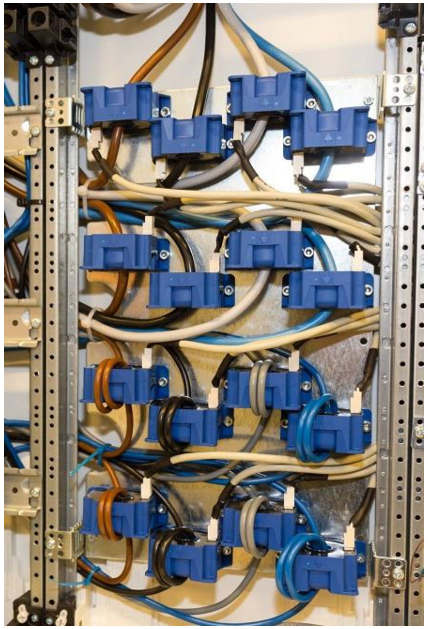
*226 current & voltage measurements, sampled at maximum 10kHz*



# Motivation – within CoSES

*How to provide accurate feedback of harmonic presence in the measurements for microgrid experiments…while not burdening the real-time controller?*

*226 current & voltage measurements, sampled at maximum 10kHz*

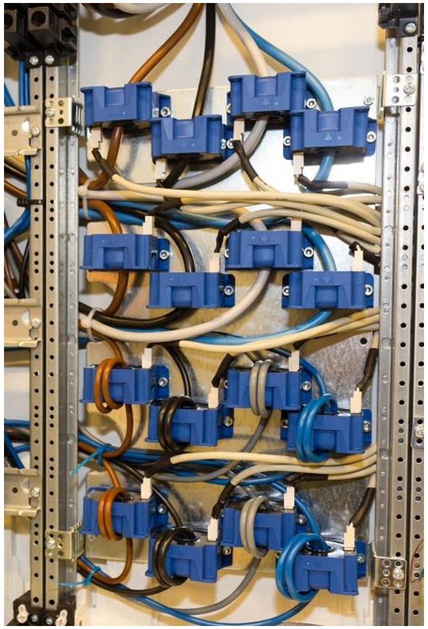


*6 embedded controllers to accquire the signals*

# Motivation – within CoSES

*How to provide accurate feedback of harmonic presence in the measurements for microgrid experiments…while not burdening the real-time controller?*

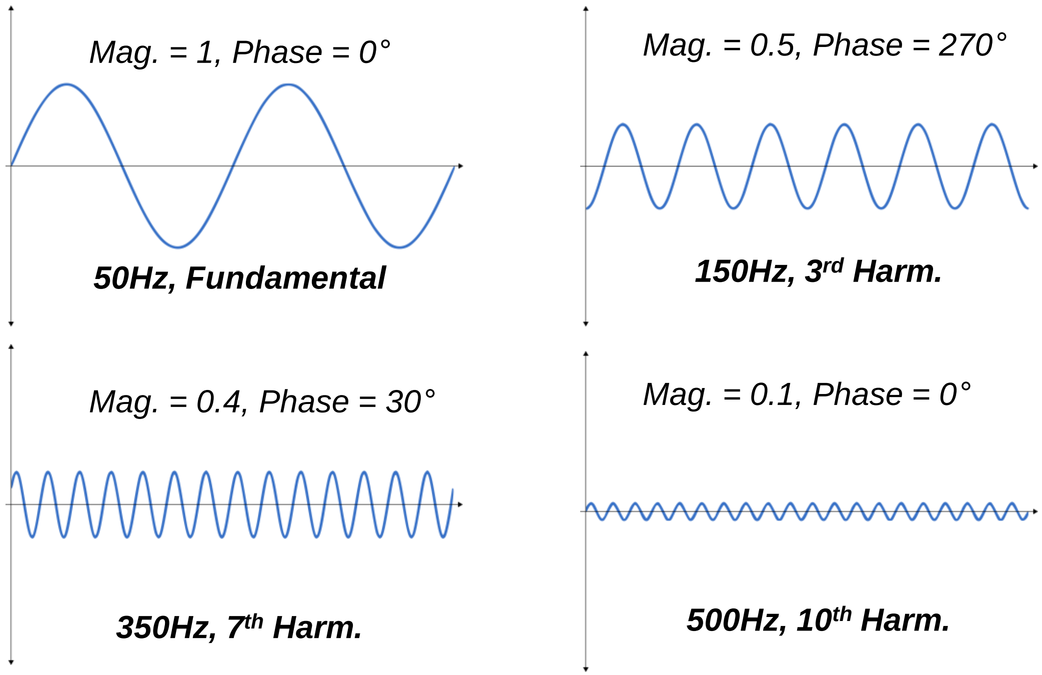
*226 current & voltage measurements, sampled at maximum 10kHz*



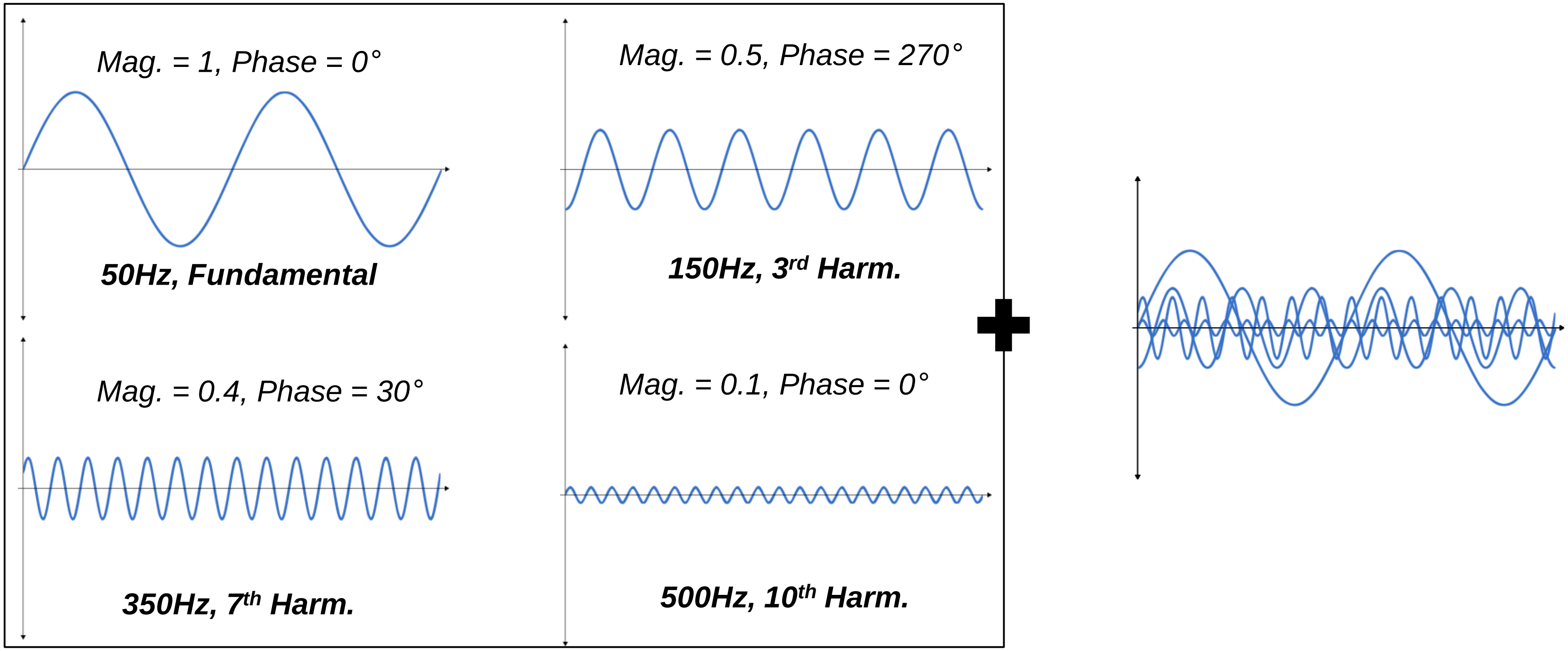
*6 embedded controllers to accquire the signals*


*Processor has many other tasks in real-time*

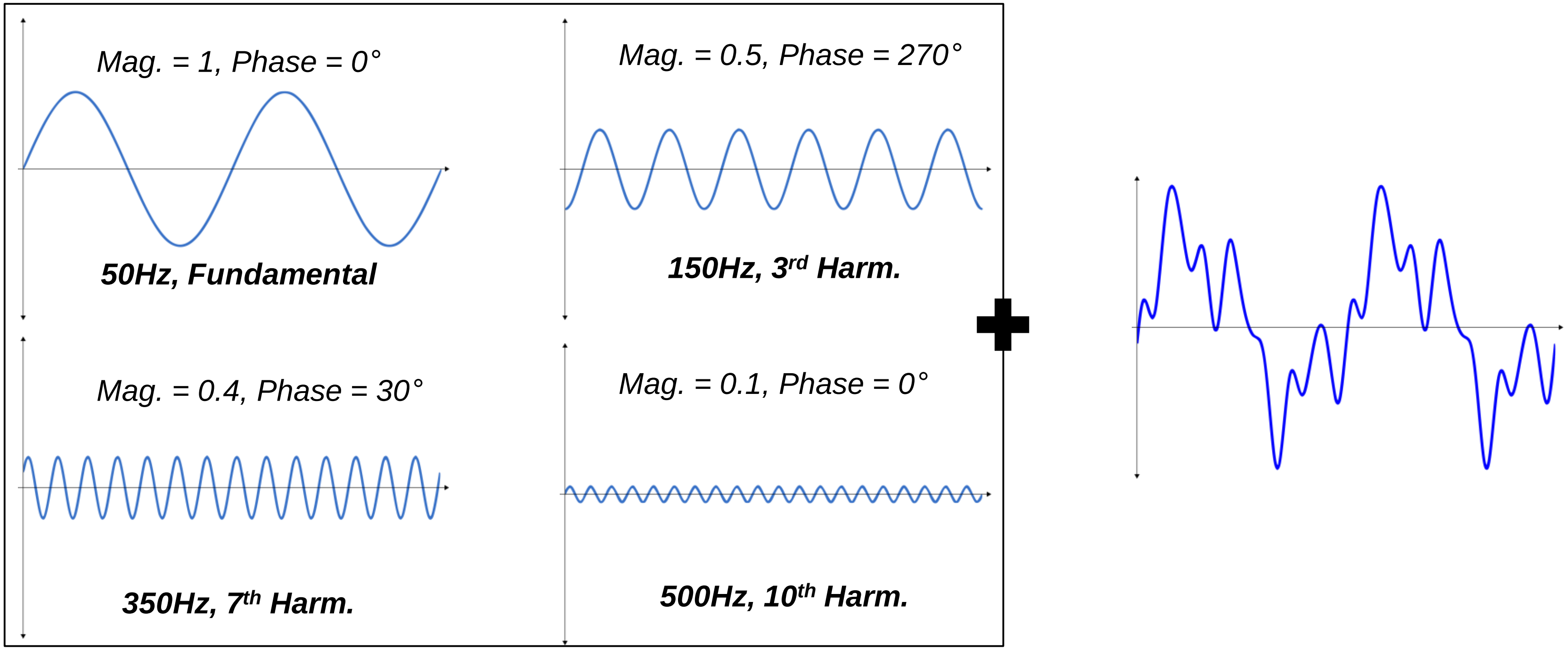
# Harmonic analysis methods



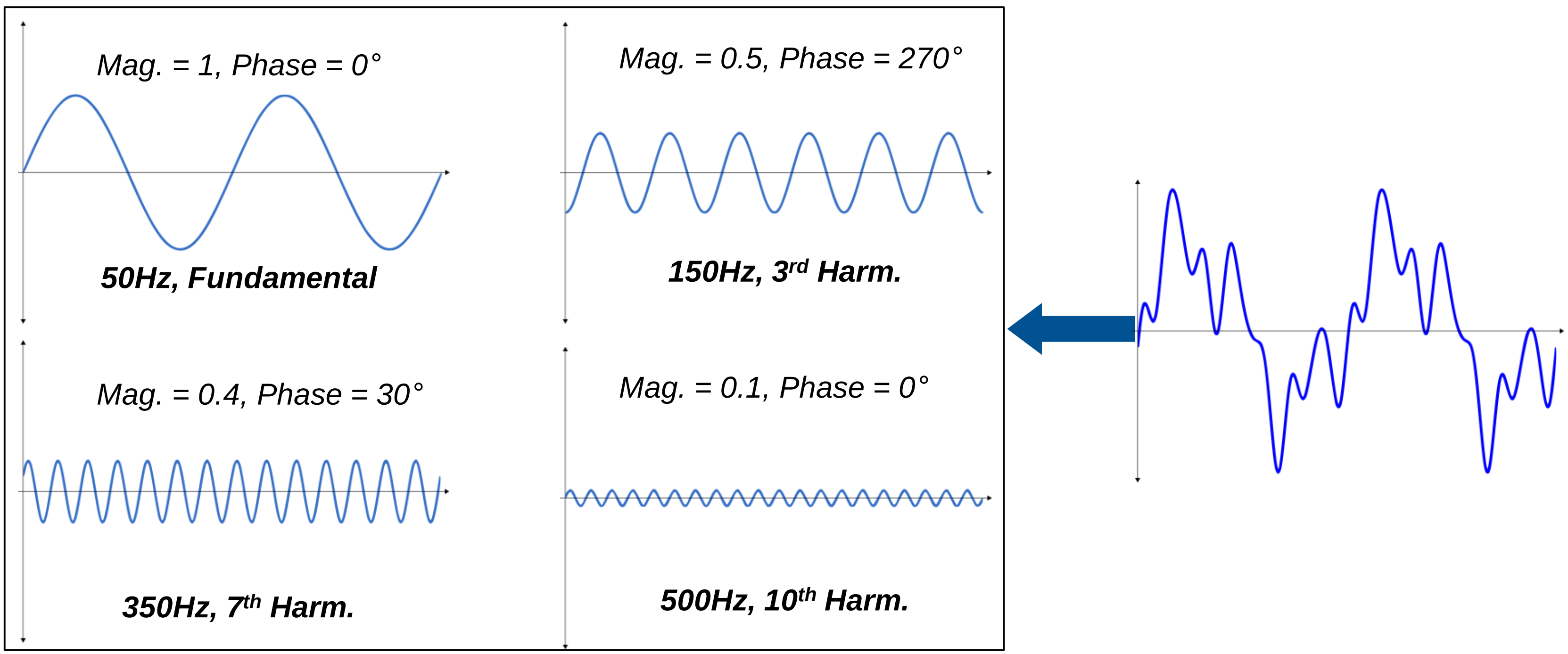
# Harmonic analysis methods



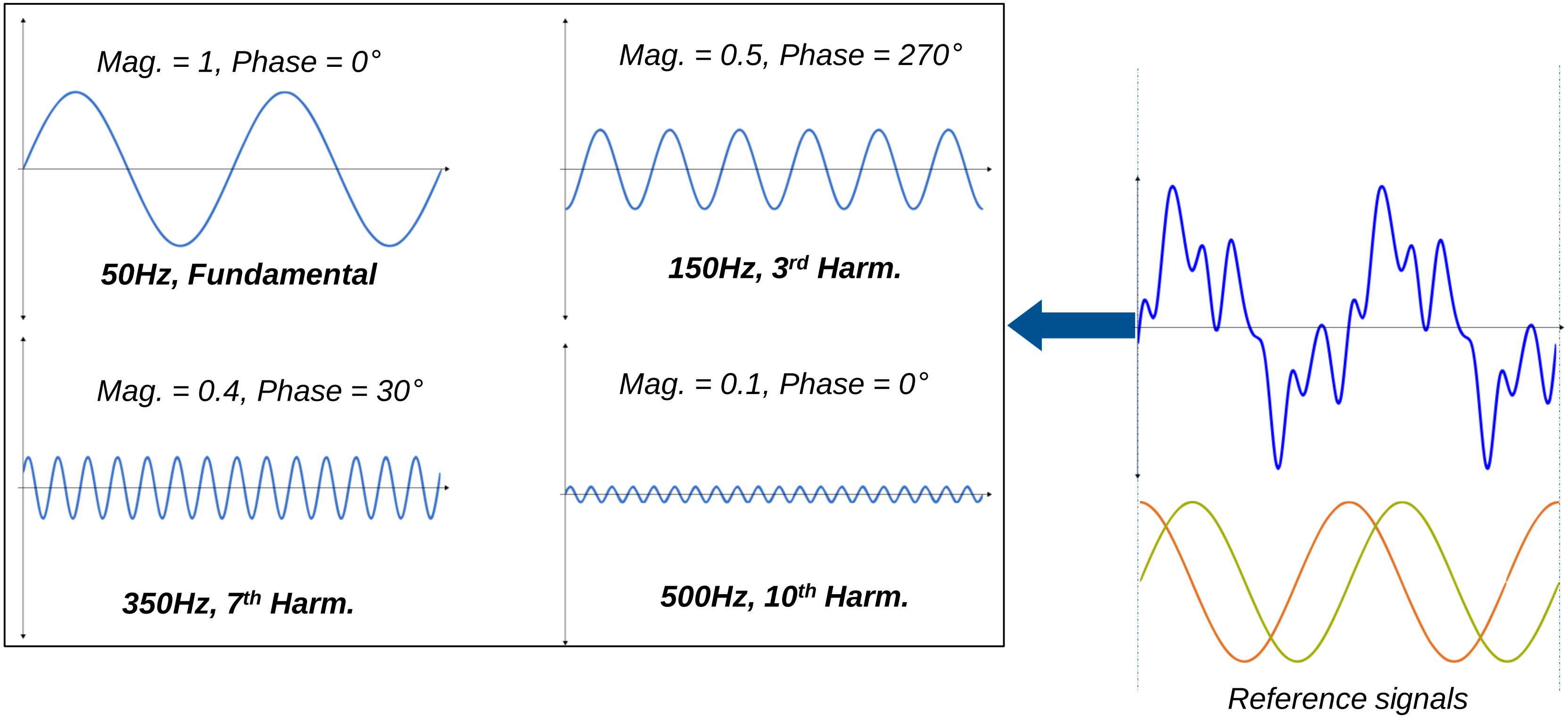
# Harmonic analysis methods



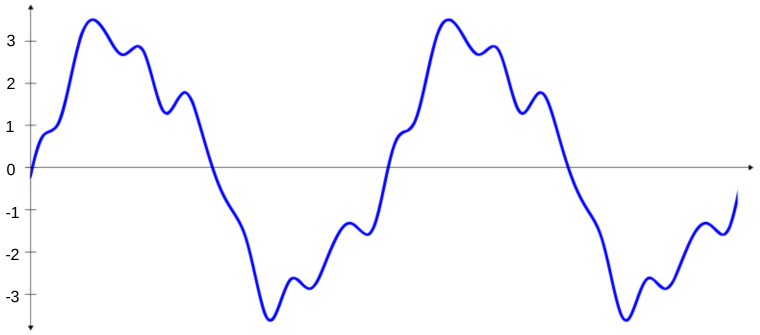
# Harmonic analysis methods



# Harmonic analysis methods



# Harmonic analysis methods



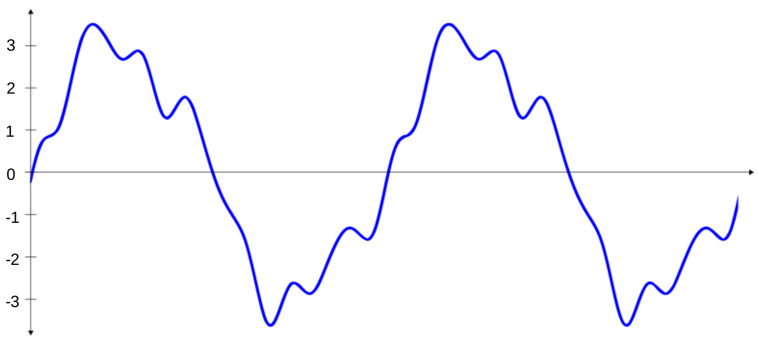

***50Hz, Mag. = 3, Phase = 0°***

*150Hz, Mag. = 0.5, Phase = 270°*

*350Hz, Mag. = 0.4, Phase = 30°*

*500Hz, Mag. = 0.1, Phase = 0°*

# Harmonic analysis methods

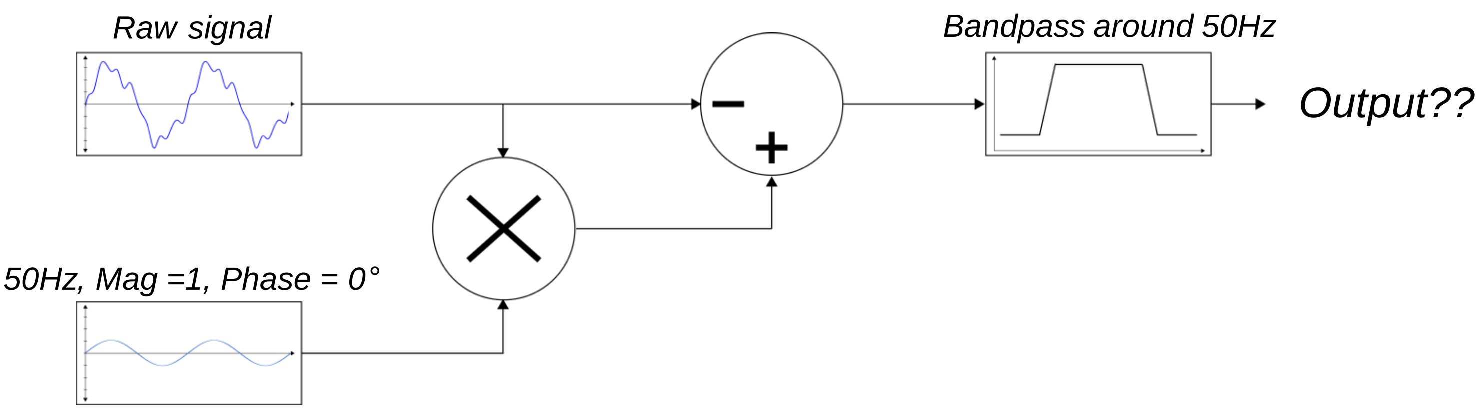



***50Hz, Mag. = 3, Phase = 0°***

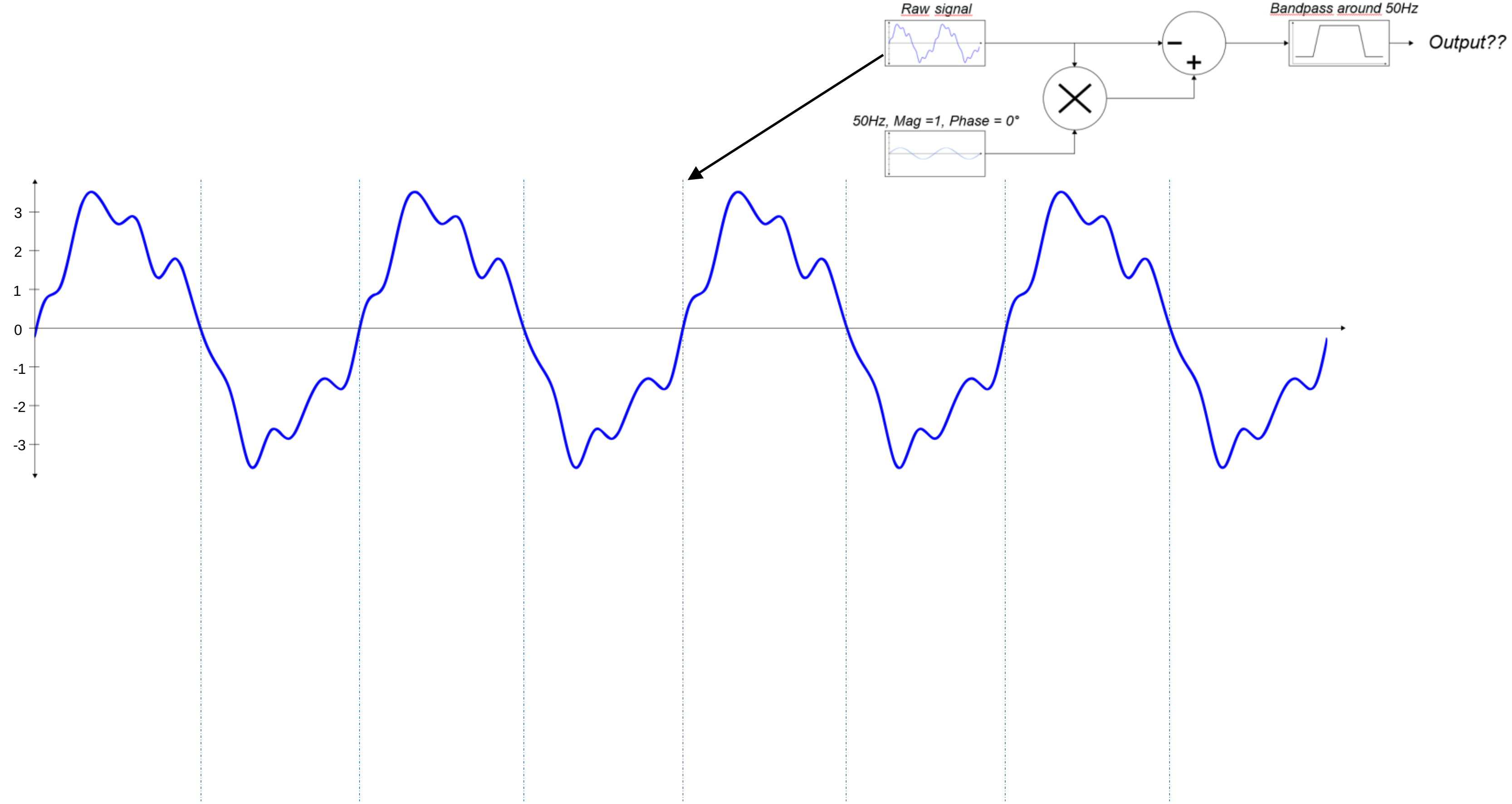
*150Hz, Mag. = 0.5, Phase = 270°*

*350Hz, Mag. = 0.4, Phase = 30°*

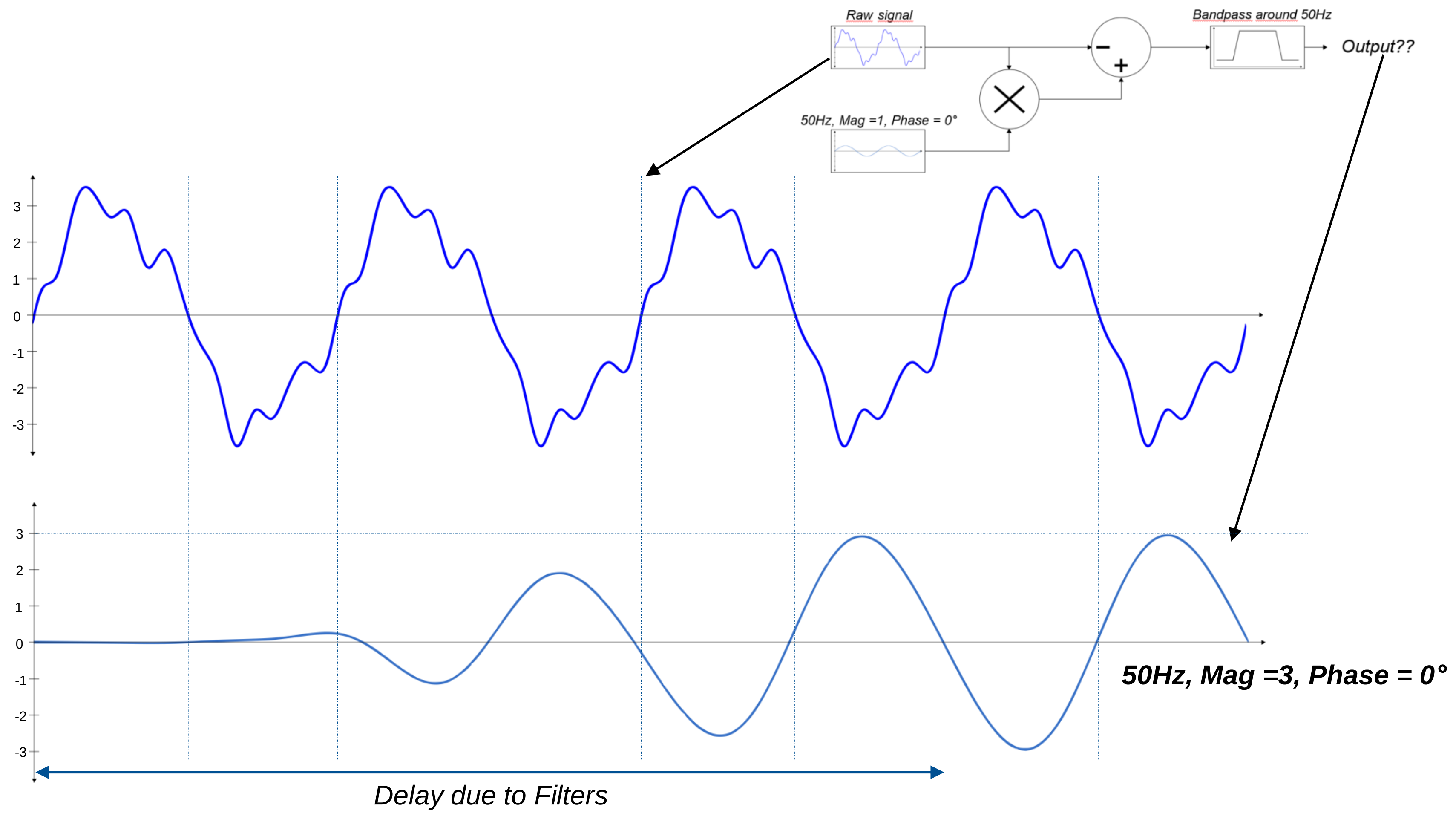
*500Hz, Mag. = 0.1, Phase = 0°*



# Harmonic analysis methods



# Harmonic analysis methods



# Harmonic analysis methods

Harmonic estimation must take into account -

**Buffer delay vs Accuracy**

**Sampling rate vs Harmonic range**

**Robust against frequency error**

**Robust against rogue harmonics**

**Sample by sample update vs Buffered frame update**

# Harmonic analysis methods

Harmonic estimation must take into account -

**Buffer delay vs Accuracy**

**Sampling rate vs Harmonic range**

**Robust against frequency error**

**Robust against rogue harmonics**

**Sample by sample update vs Buffered frame update**

**Calculations must finish within iteration of the real-time loop**

# Harmonic analysis methods

Fourier based methods

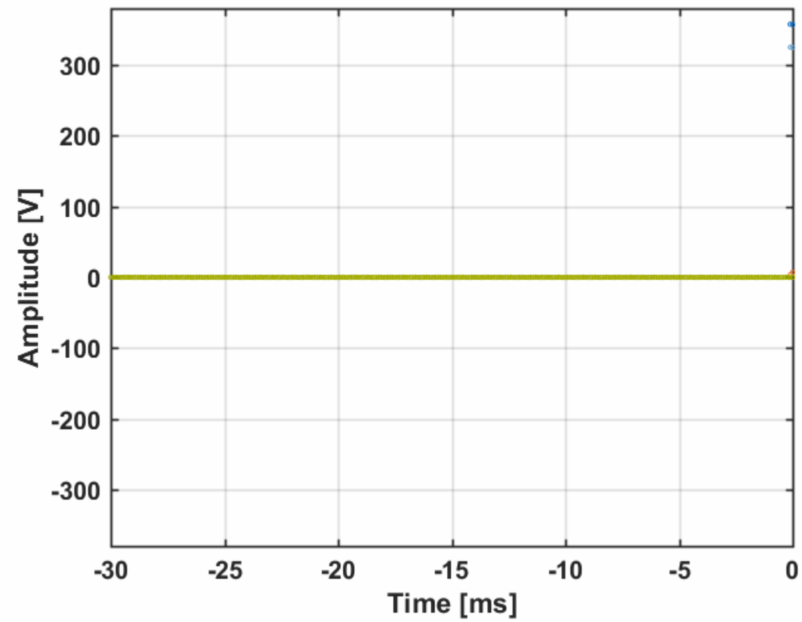
- Discrete fourier transform (DFT)
  - Sliding Discrete Fourier Transform (**SDFT**)
  - Modulated Sliding Discrete Fourier Transform (**mSDFT**)

Time-frequency based methods

- Second Order Generalized Integrators (**SOGI**)

# Real time processing on HIL targets

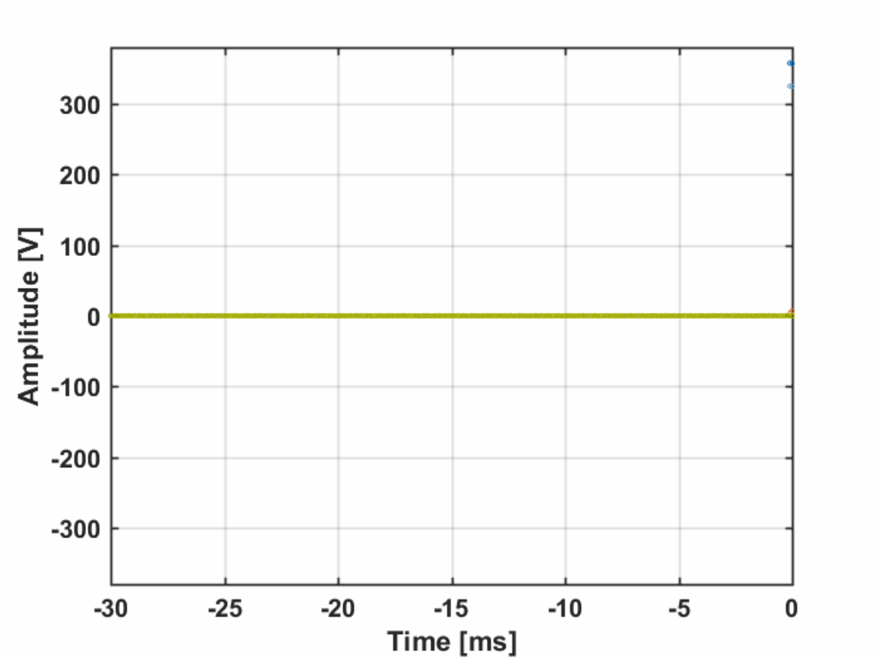
- Maximum sampling rate is **10kHz**
- On each iteration, all of the calculations in the model should be completed to prevent data loss.
- Otherwise, the controller either **skips** the model (*overrun*), or **cannot react** real time (*latency*).




*Frame based processing vs sample by sample processing. input, fundamental component, sample by sample (SDFT), frame based (DFT).*

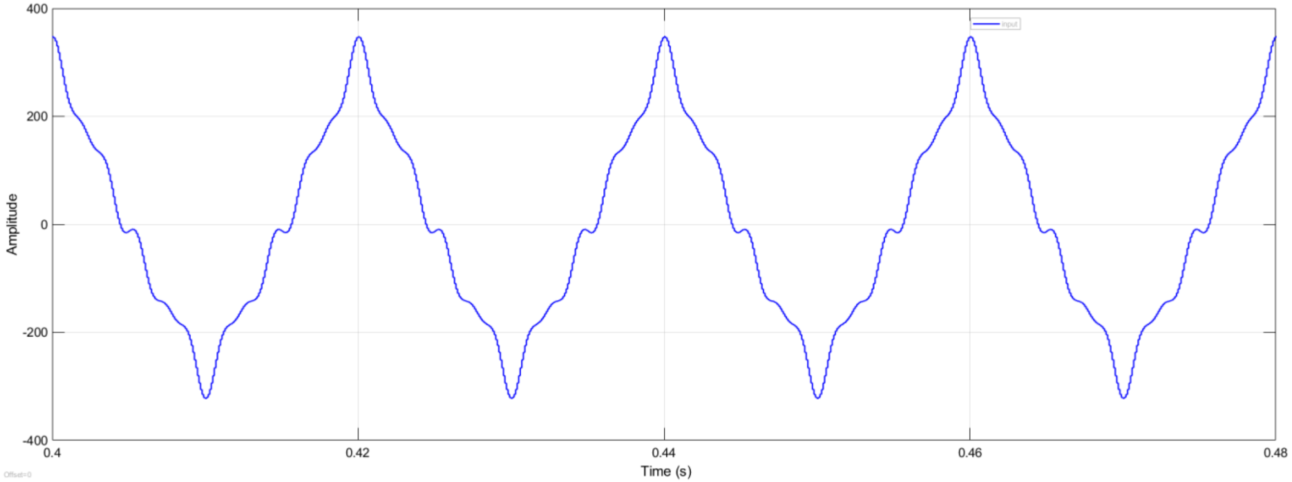
# Real time processing on HIL targets

- Maximum sampling rate is **10kHz**
- On each iteration, all of the calculations in the model should be completed to prevent data loss.
- Otherwise, the controller either **skips** the model (*overrun*), or **cannot react** real time (*latency*).




*Frame based processing vs sample by sample processing. input, fundamental component, sample by sample (SDFT), frame based (DFT).*

# Real time processing on HIL targets

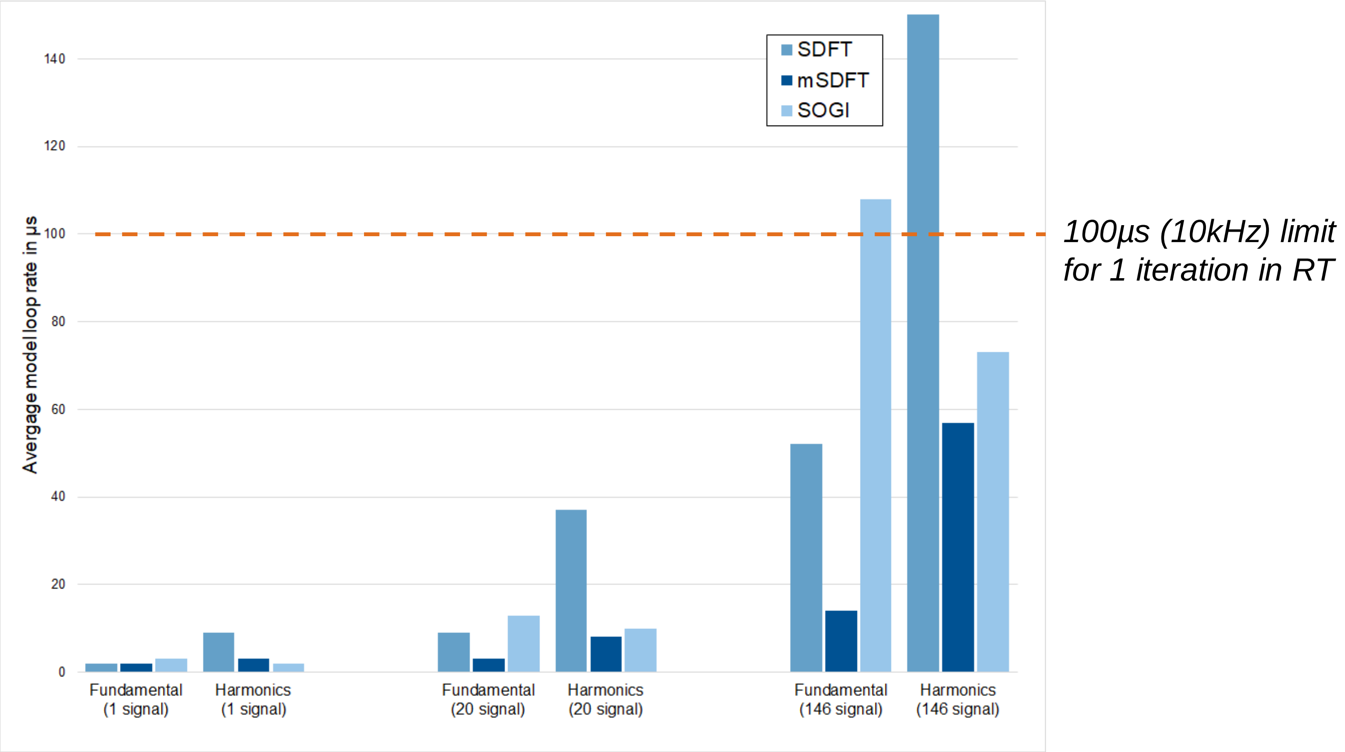



*Input signal to test the real time harmonic estimation*

*Harmonic orders –* ***$1^{st}$, $2^{nd}$, $3^{rd}$, $5^{th}$, $7^{th}$,$11^{th}$, $13^{th}$***

# Real time processing on HIL targets

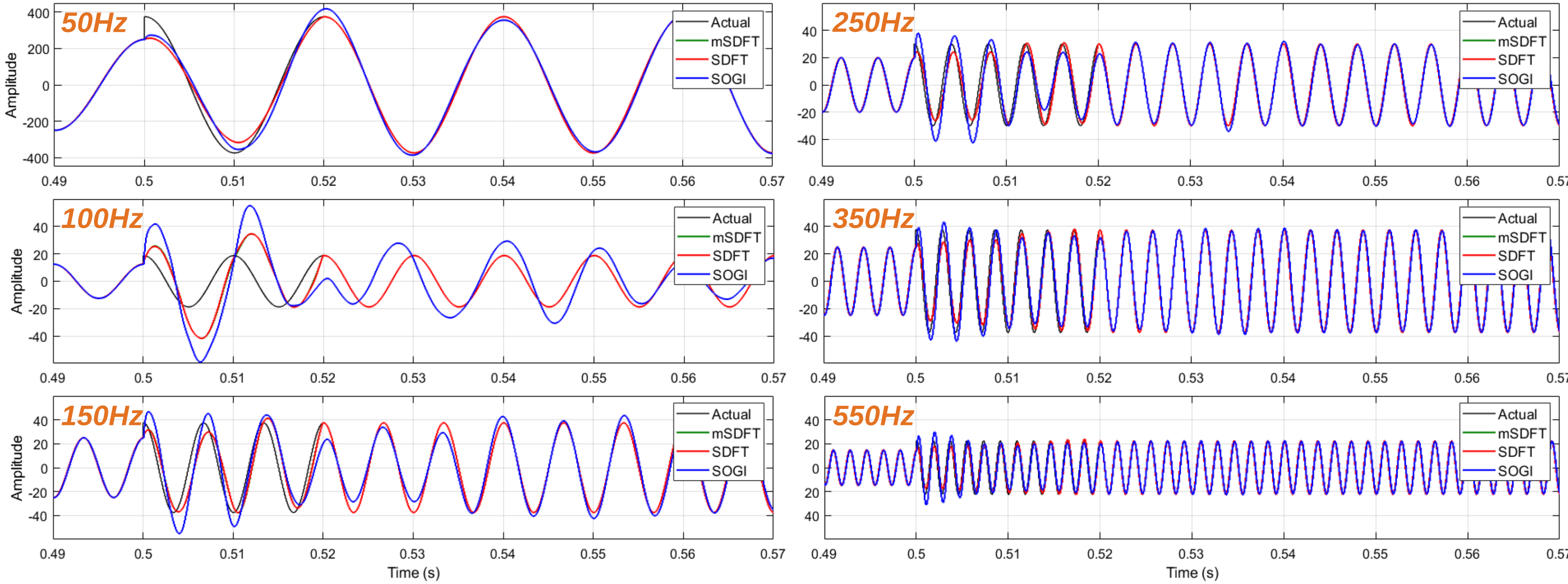
- Outputs are estimated waveforms; not just magnitudes with phase angles.




*RT computation time results for SDFT, mSDFT and SOGI*

# Real time processing on HIL targets

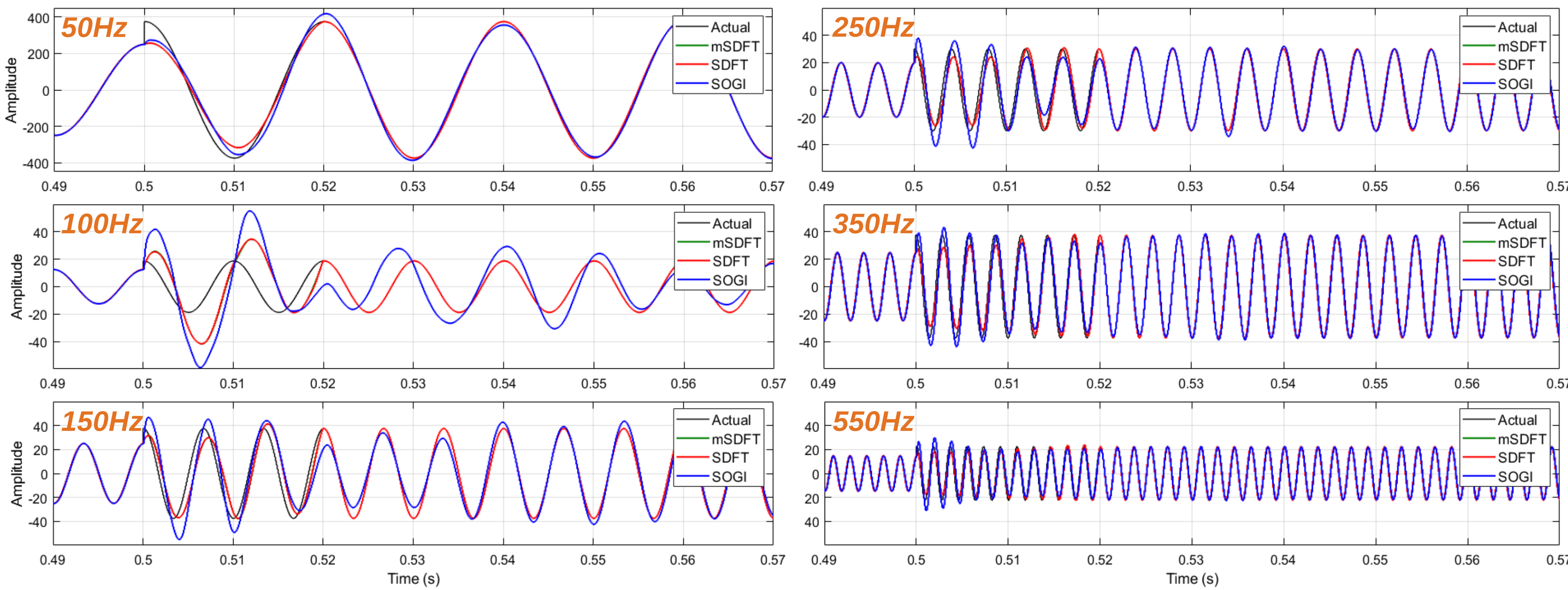
- Outputs are estimated waveforms; not just magnitudes with phase angles.




*RT test results step change at 0.5s for SDFT, mSDFT and SOGI*

# Real time processing on HIL targets

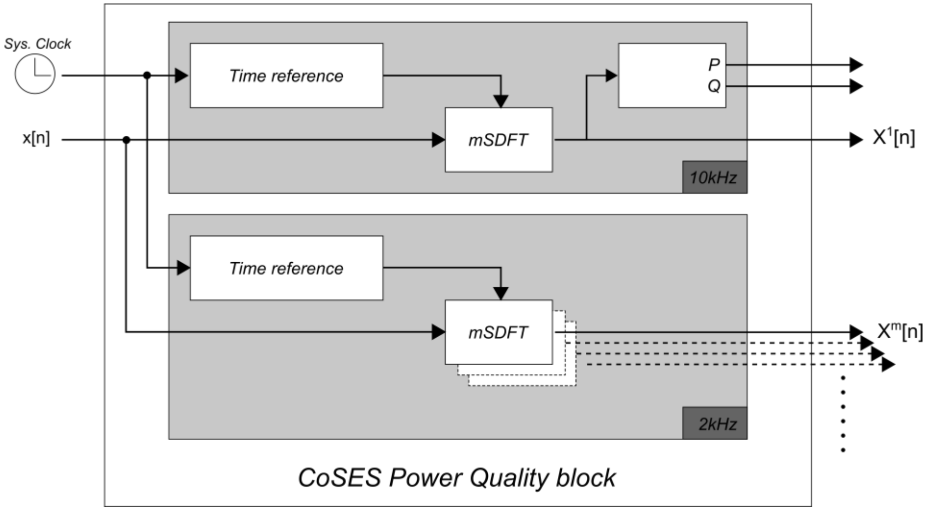
- Outputs are estimated waveforms; not just magnitudes with phase angles.
- **mSDFT** outperformed the other methods in **less burden** and **equally accurate** results.




*RT test results step change at 0.5s for SDFT, mSDFT and SOGI*

# CoSES PQ meter using mSDFT

- A compromise is made,
  - Fundamental components at **10kHz**
  - Harmonic components at **2kHz**
- It is possible to process even 146 measurements on a single PXI and obtain phasors of the entire grid instantaneously.
- While still leaving considerable processing power for other tasks.



# Conclusions and future work

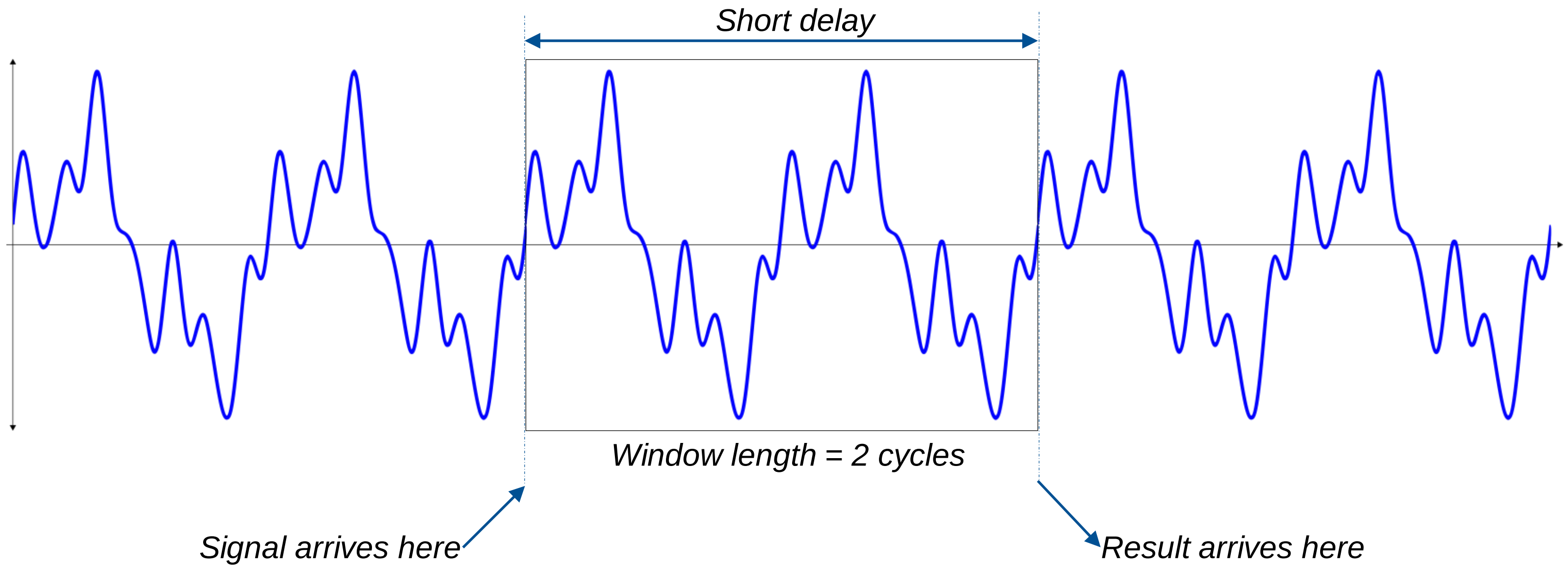
- Various signal processing algorithms and data processing approaches are compared for monitoring of CoSES
- An efficient, fast and robust PQ meter application using mSDFT has been implemented
- PQ meter has possibility to reduce controller burden even more by switching to buffered frame updates if necessary
- Future work
  - Reducing frequency fluctuation dependency.
  - Using the PQ meter for feedback in a live experiment.

# Supplemenatary Slides

# Implementation nuances

## Fourier based methods

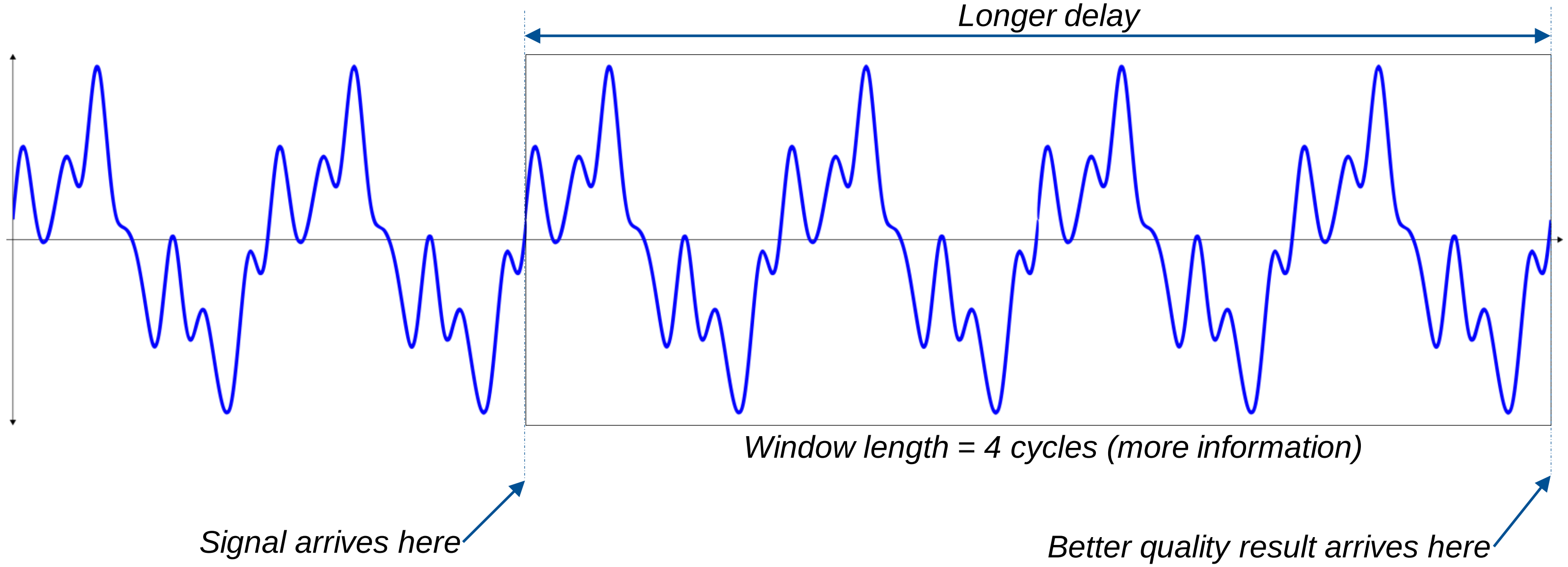
- Discrete fourier transform (DFT)
- Sliding Discrete Fourier Transform (SDFT)
- Modulated Sliding Discrete Fourier Transform (mSDFT)



# Implementation nuances

## Fourier based methods

- Discrete fourier transform (DFT)
- Sliding Discrete Fourier Transform (SDFT)
- Modulated Sliding Discrete Fourier Transform (mSDFT)

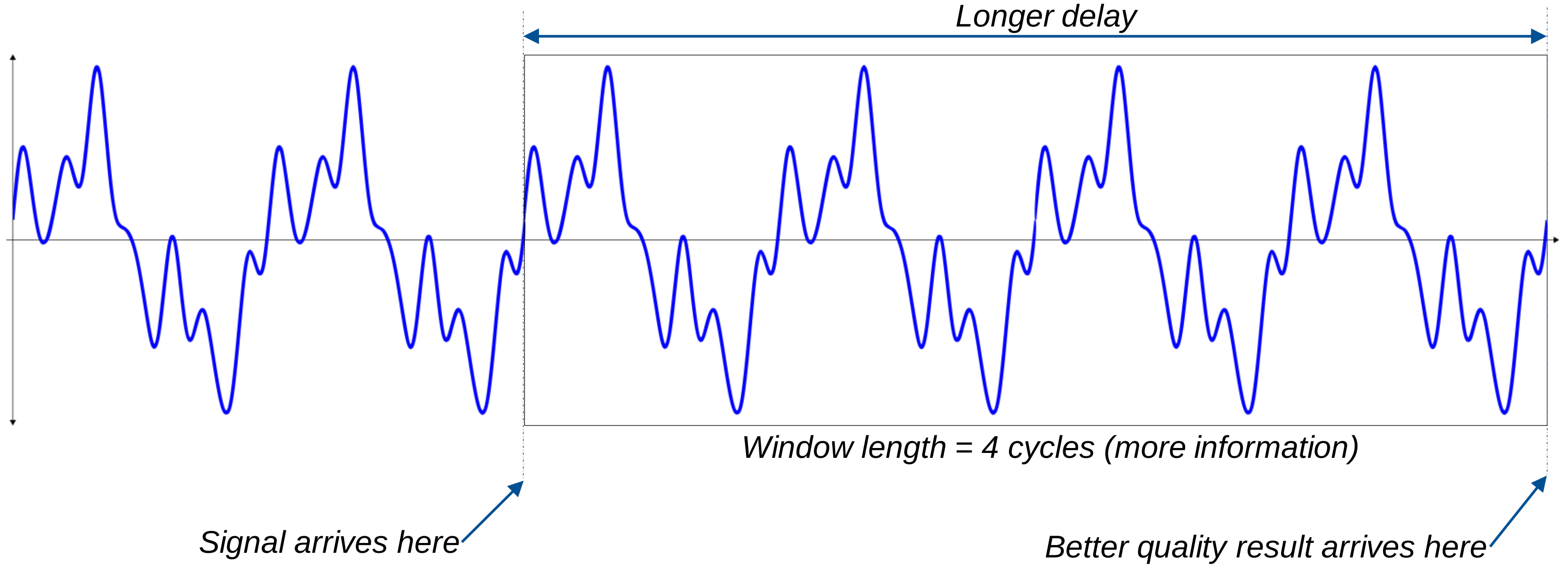


# Implementation nuances

## Fourier based methods

- Discrete fourier transform (DFT)
- Sliding Discrete Fourier Transform (SDFT)
- Modulated Sliding Discrete Fourier Transform (mSDFT)

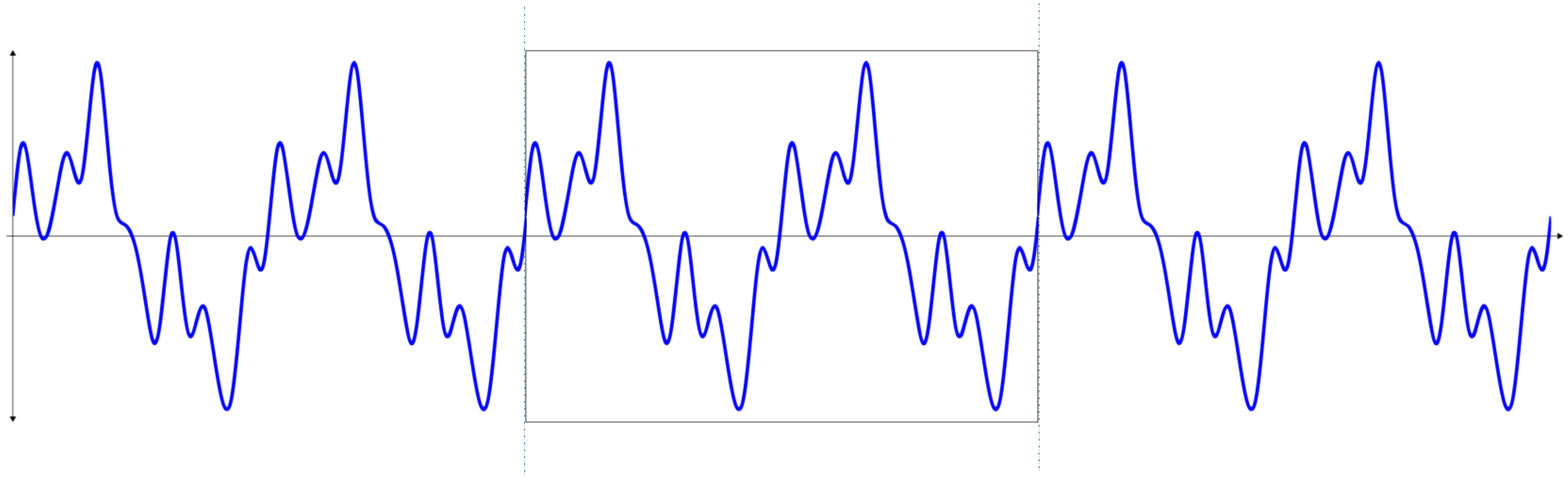
**Buffer delay vs Accuracy**



# Implementation nuances

## Fourier based methods

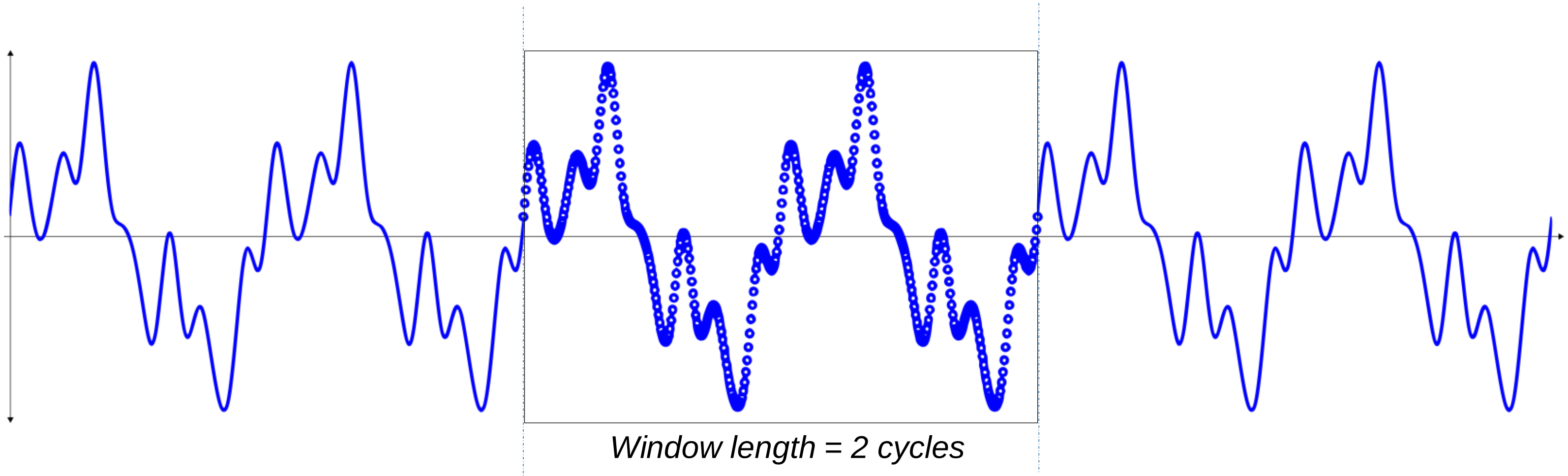
- Discrete fourier transform (DFT)
- Sliding Discrete Fourier Transform (SDFT)
- Modulated Sliding Discrete Fourier Transform (mSDFT)



# Implementation nuances

## Fourier based methods

- Discrete fourier transform (DFT)
- Sliding Discrete Fourier Transform (SDFT)
- Modulated Sliding Discrete Fourier Transform (mSDFT)

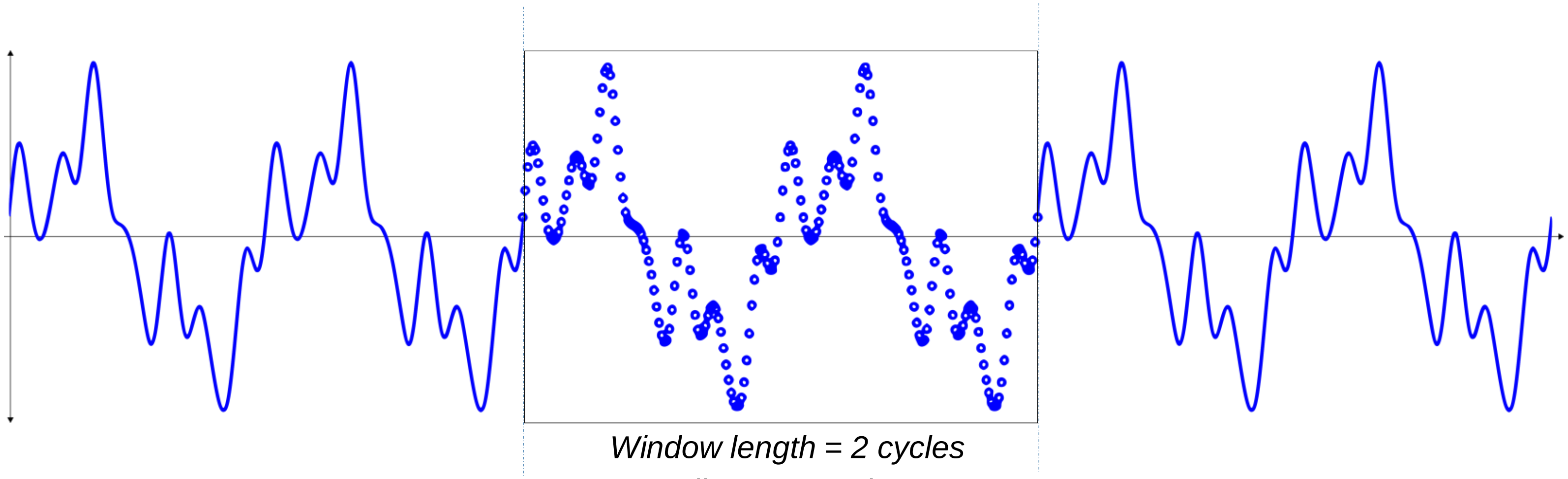


*Window length = 2 cycles*
*Sampling rate = 10 kHz*
*Computation rate High*

# Implementation nuances

## Fourier based methods

- Discrete fourier transform (DFT)
- Sliding Discrete Fourier Transform (SDFT)
- Modulated Sliding Discrete Fourier Transform (mSDFT)

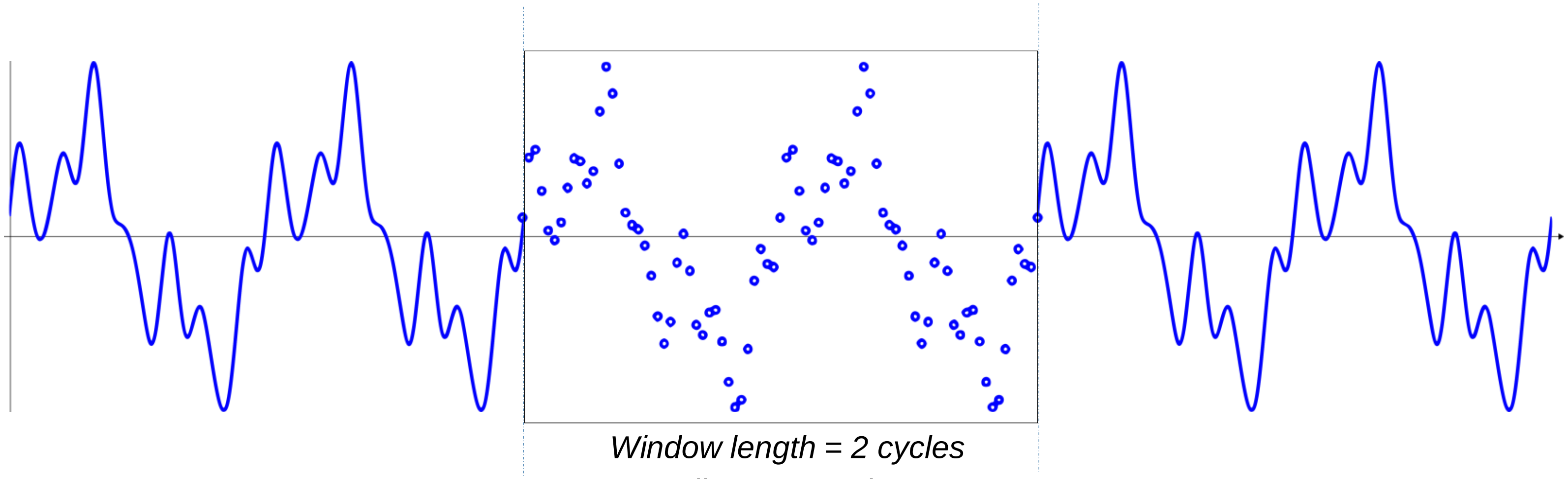


*Window length = 2 cycles*
*Sampling rate = 5 kHz*
*Computation rate medium*

# Implementation nuances

## Fourier based methods

- Discrete fourier transform (DFT)
- Sliding Discrete Fourier Transform (SDFT)
- Modulated Sliding Discrete Fourier Transform (mSDFT)

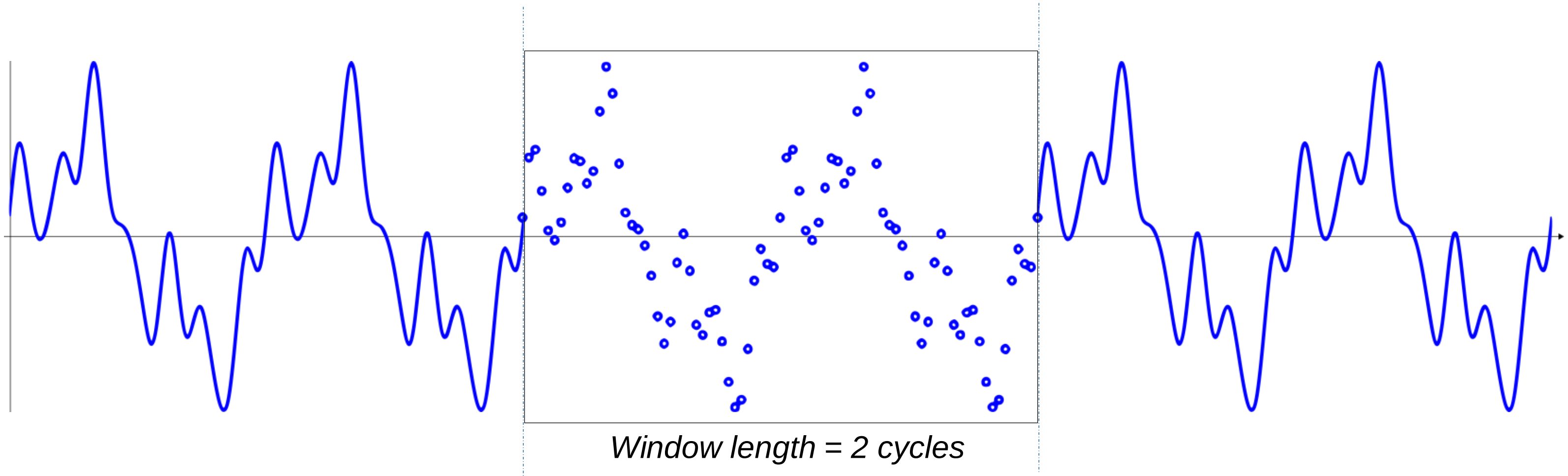


*Window length = 2 cycles*
*Sampling rate = 2 kHz*
*Computation rate low*

# Implementation nuances

## Fourier based methods

- Discrete fourier transform (DFT)
- Sliding Discrete Fourier Transform (SDFT)
- Modulated Sliding Discrete Fourier Transform (mSDFT)



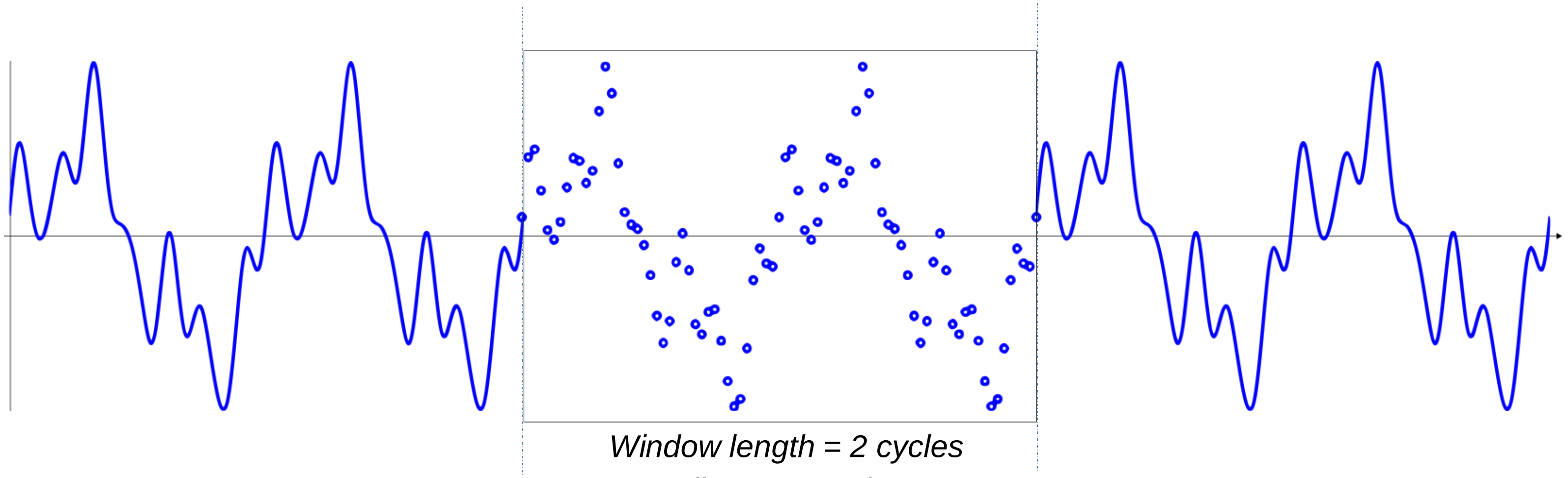
*Window length = 2 cycles*
*Sampling rate = 2 kHz*
*Computation rate low*

# Implementation nuances

## Fourier based methods

- Discrete fourier transform (DFT)
- Sliding Discrete Fourier Transform (SDFT)
- Modulated Sliding Discrete Fourier Transform (mSDFT)

**Sampling rate vs Harmonic range**

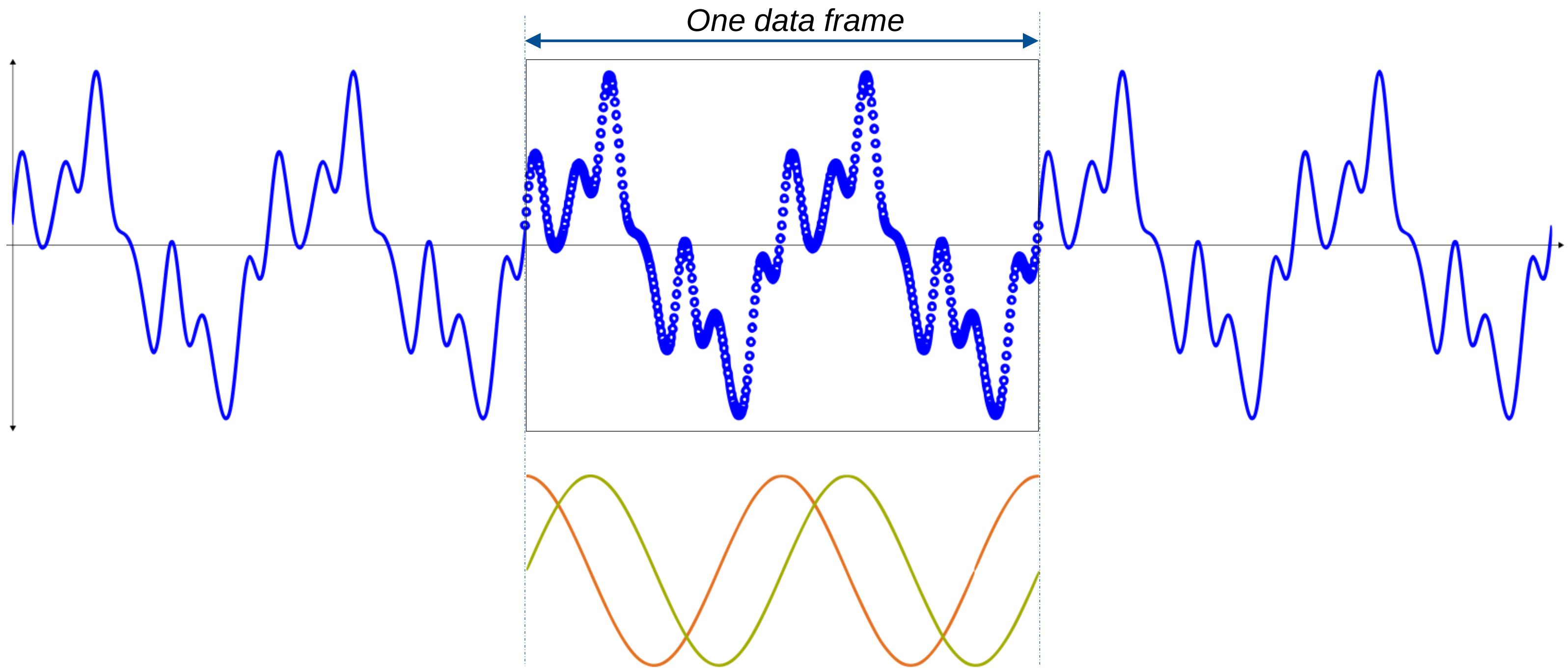


*Window length = 2 cycles*
*Sampling rate = 2 kHz*
*Computation rate low*

# Implementation nuances

## Fourier based methods

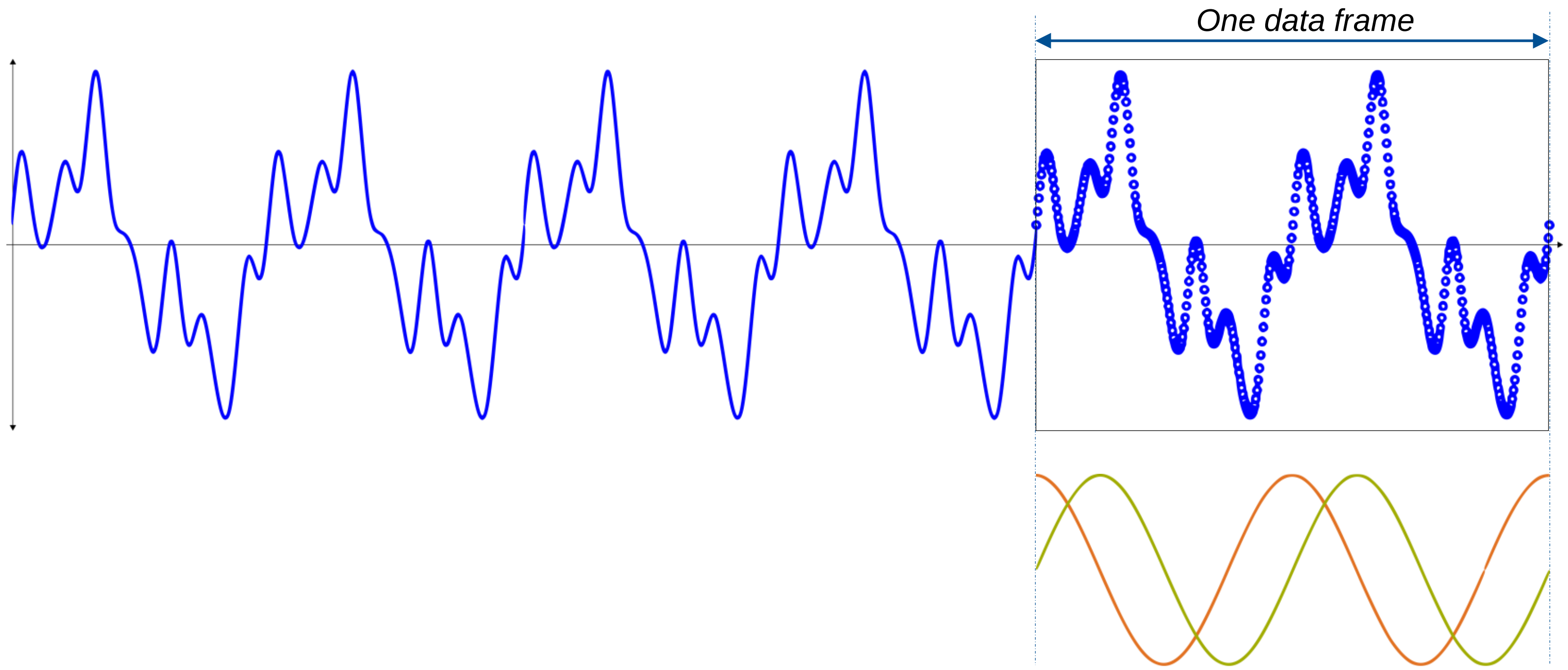
- Discrete fourier transform (DFT)
- Sliding Discrete Fourier Transform (SDFT)
- Modulated Sliding Discrete Fourier Transform (mSDFT)



# Implementation nuances

## Fourier based methods

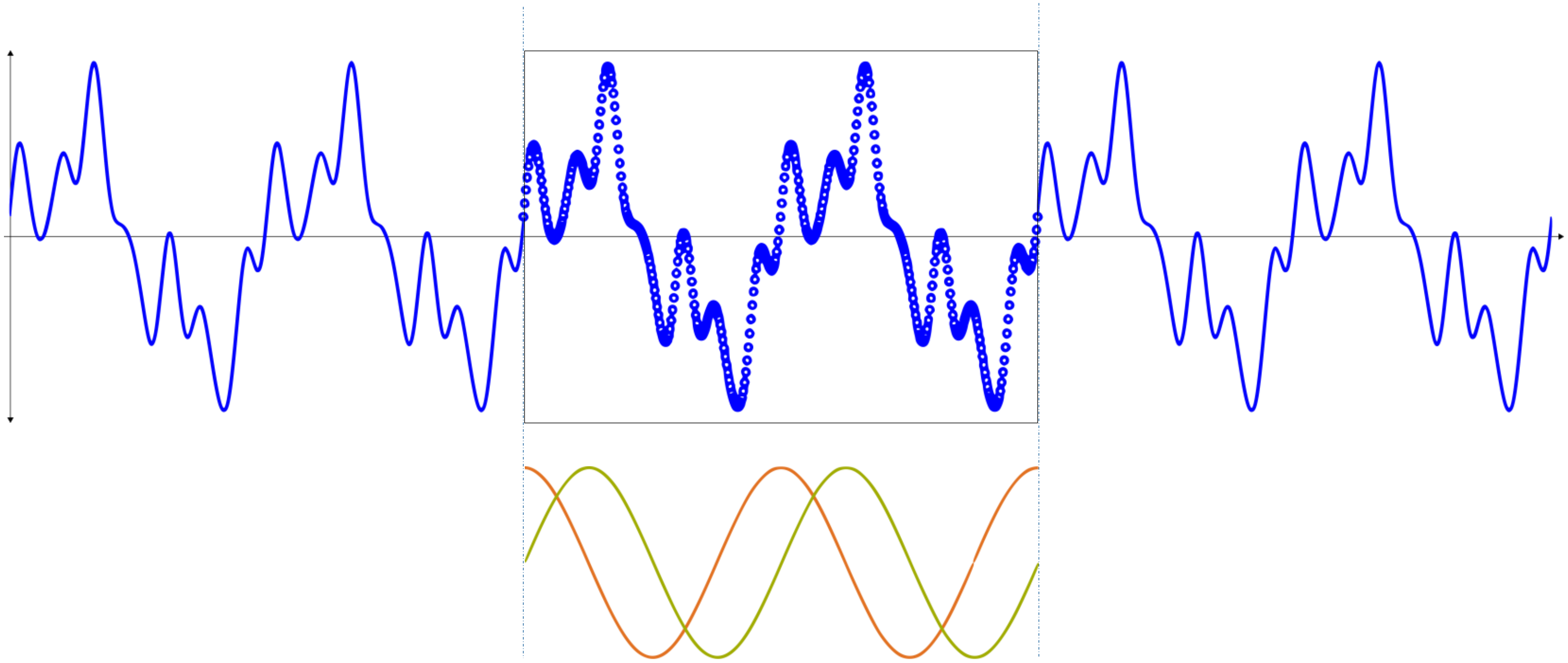
- Discrete fourier transform (DFT)
- Sliding Discrete Fourier Transform (SDFT)
- Modulated Sliding Discrete Fourier Transform (mSDFT)



# Implementation nuances

## Fourier based methods

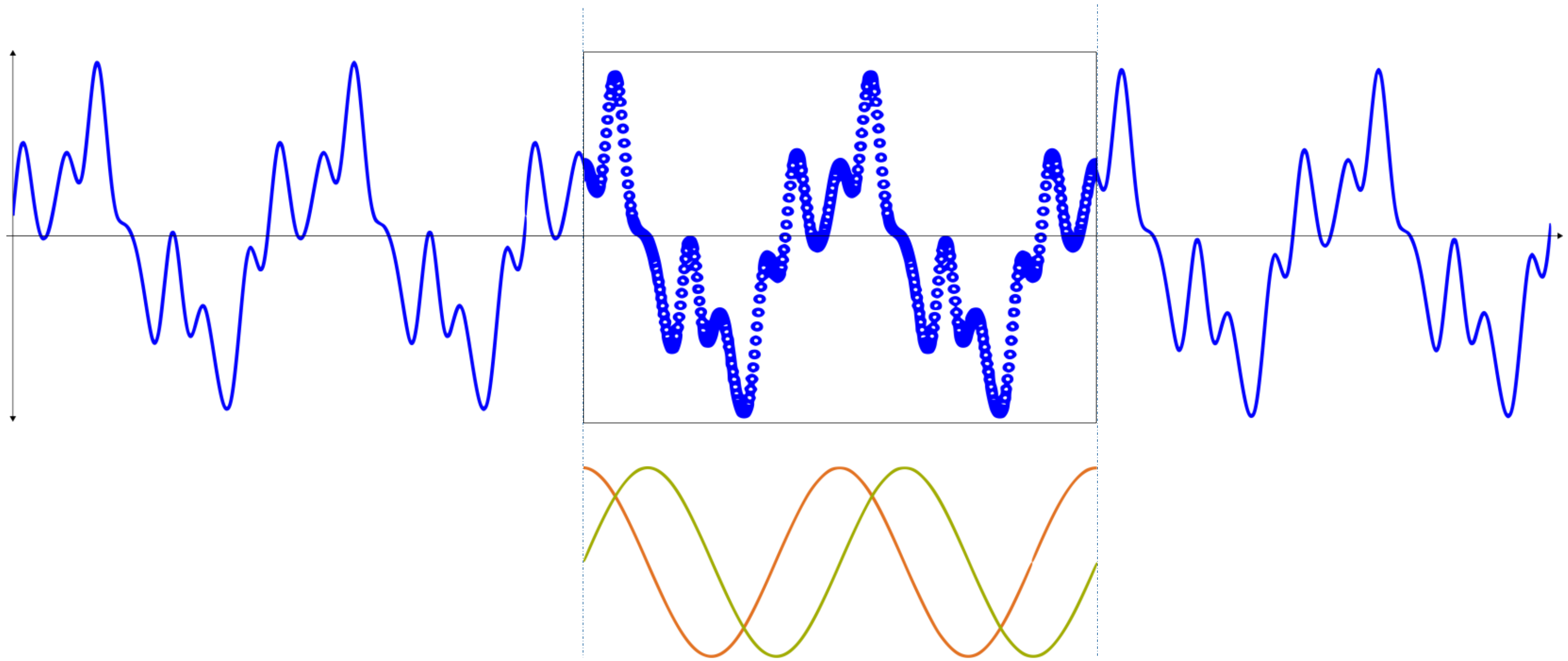
- Discrete fourier transform (DFT)
- Sliding Discrete Fourier Transform (SDFT)
- Modulated Sliding Discrete Fourier Transform (mSDFT)



# Implementation nuances

## Fourier based methods

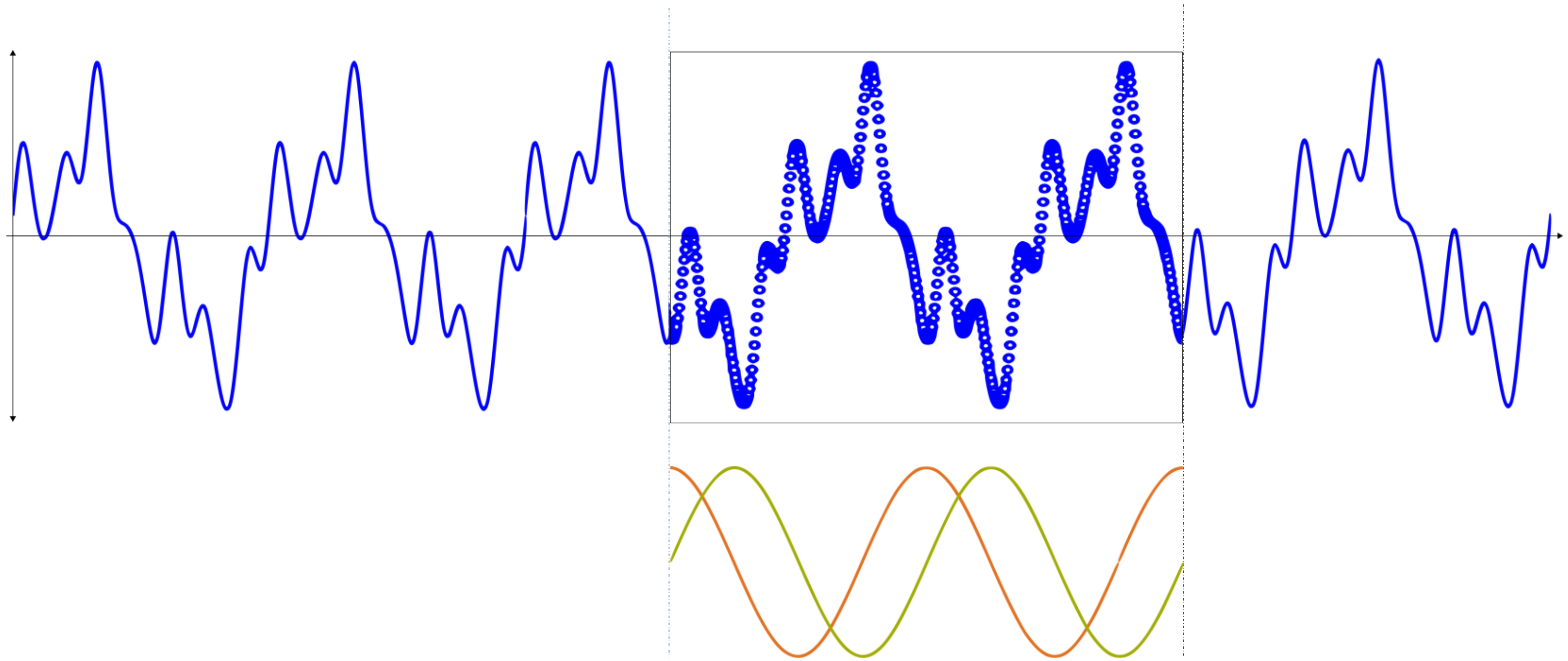
- Discrete fourier transform (DFT)
- Sliding Discrete Fourier Transform (SDFT)
- Modulated Sliding Discrete Fourier Transform (mSDFT)



# Implementation nuances

## Fourier based methods

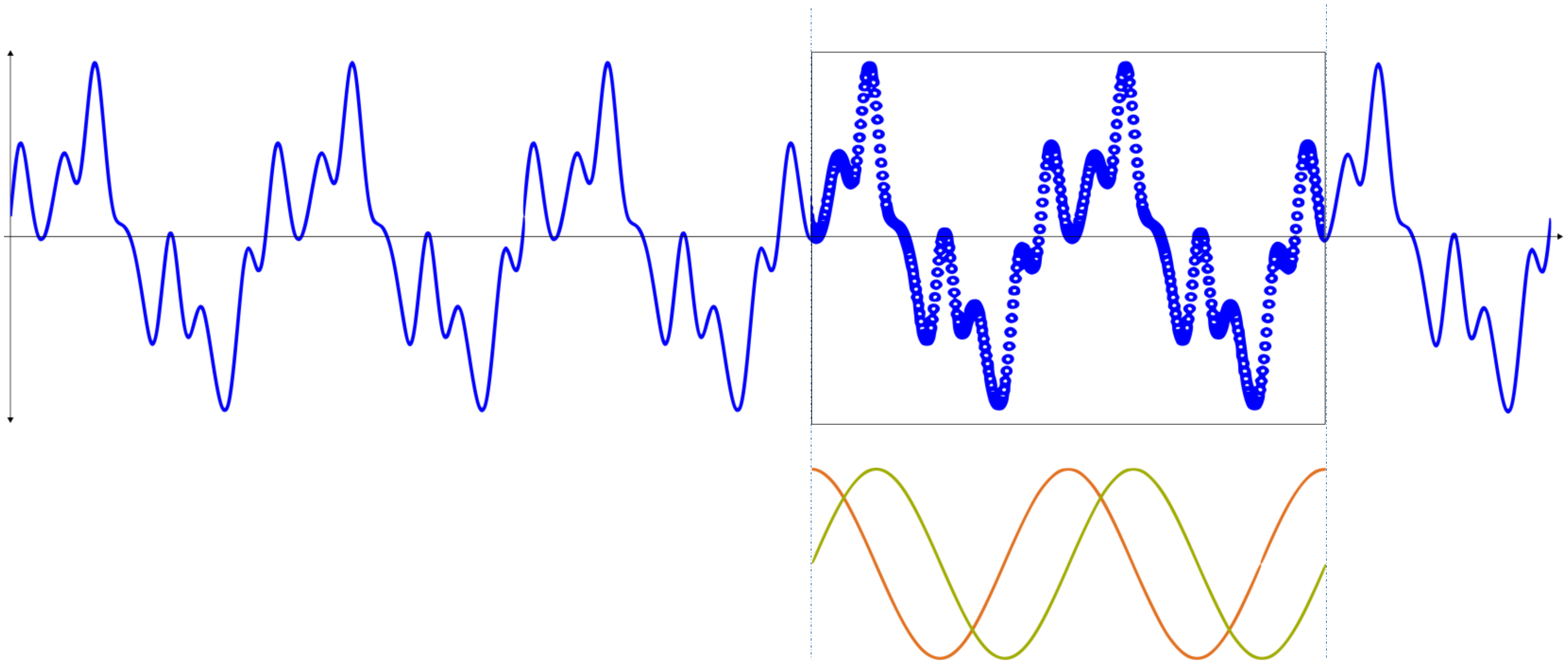
- Discrete fourier transform (DFT)
- Sliding Discrete Fourier Transform (SDFT)
- Modulated Sliding Discrete Fourier Transform (mSDFT)



# Implementation nuances

## Fourier based methods

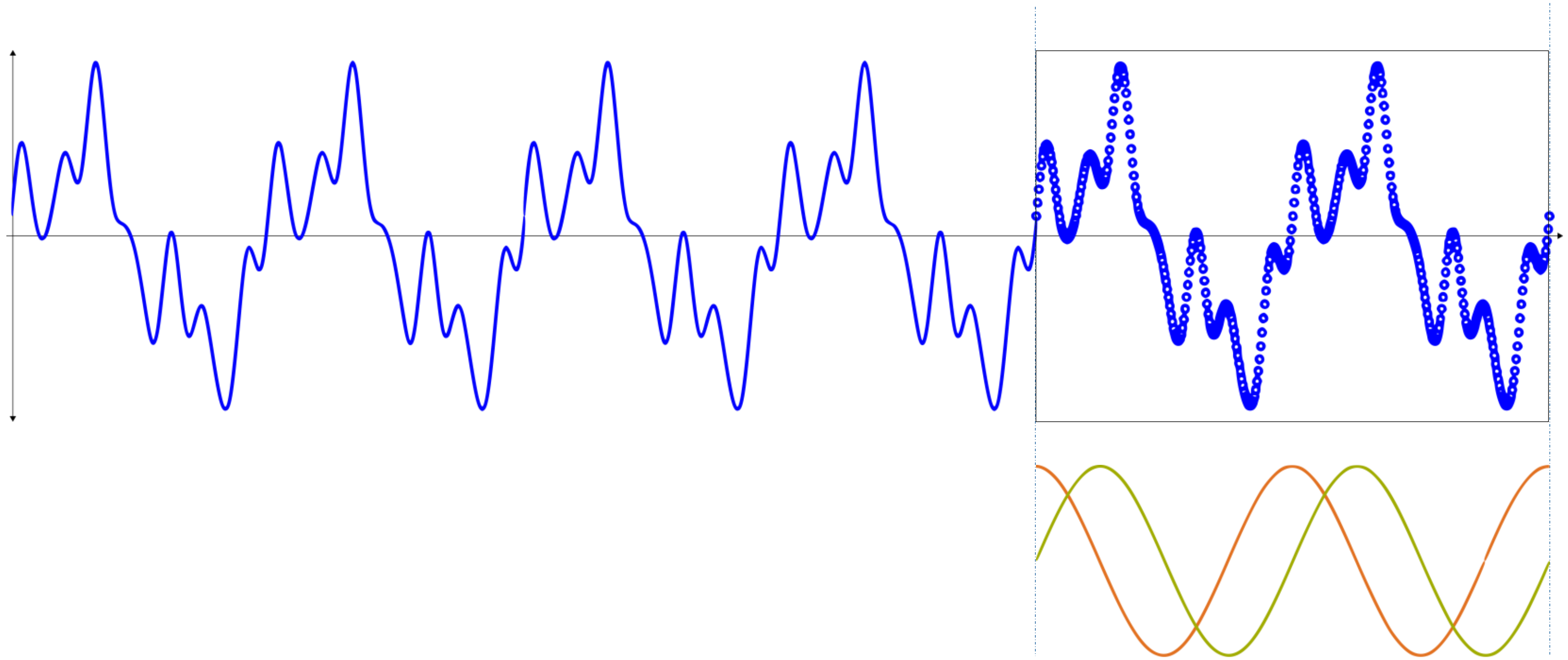
- Discrete fourier transform (DFT)
- Sliding Discrete Fourier Transform (SDFT)
- Modulated Sliding Discrete Fourier Transform (mSDFT)



# Implementation nuances

## Fourier based methods

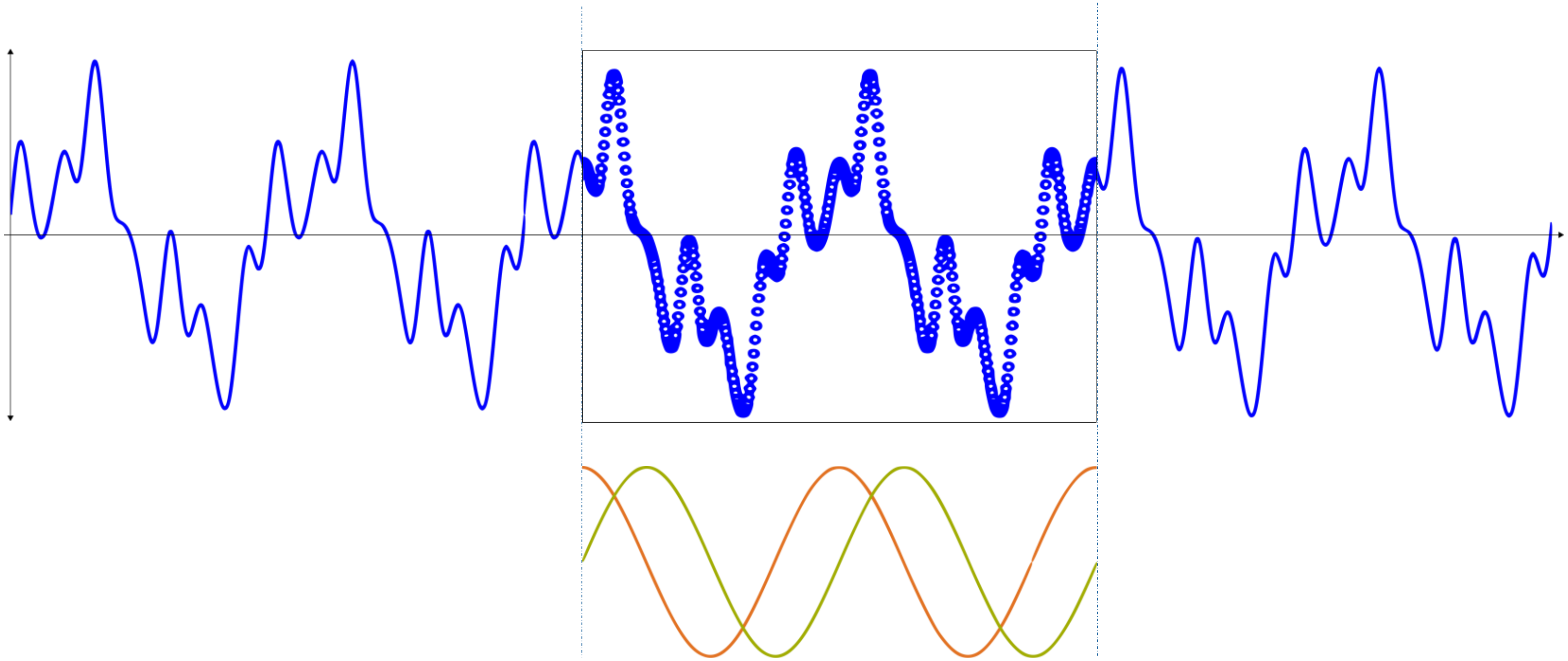
- Discrete fourier transform (DFT)
- Sliding Discrete Fourier Transform (SDFT)
- Modulated Sliding Discrete Fourier Transform (mSDFT)



# Implementation nuances

## Fourier based methods

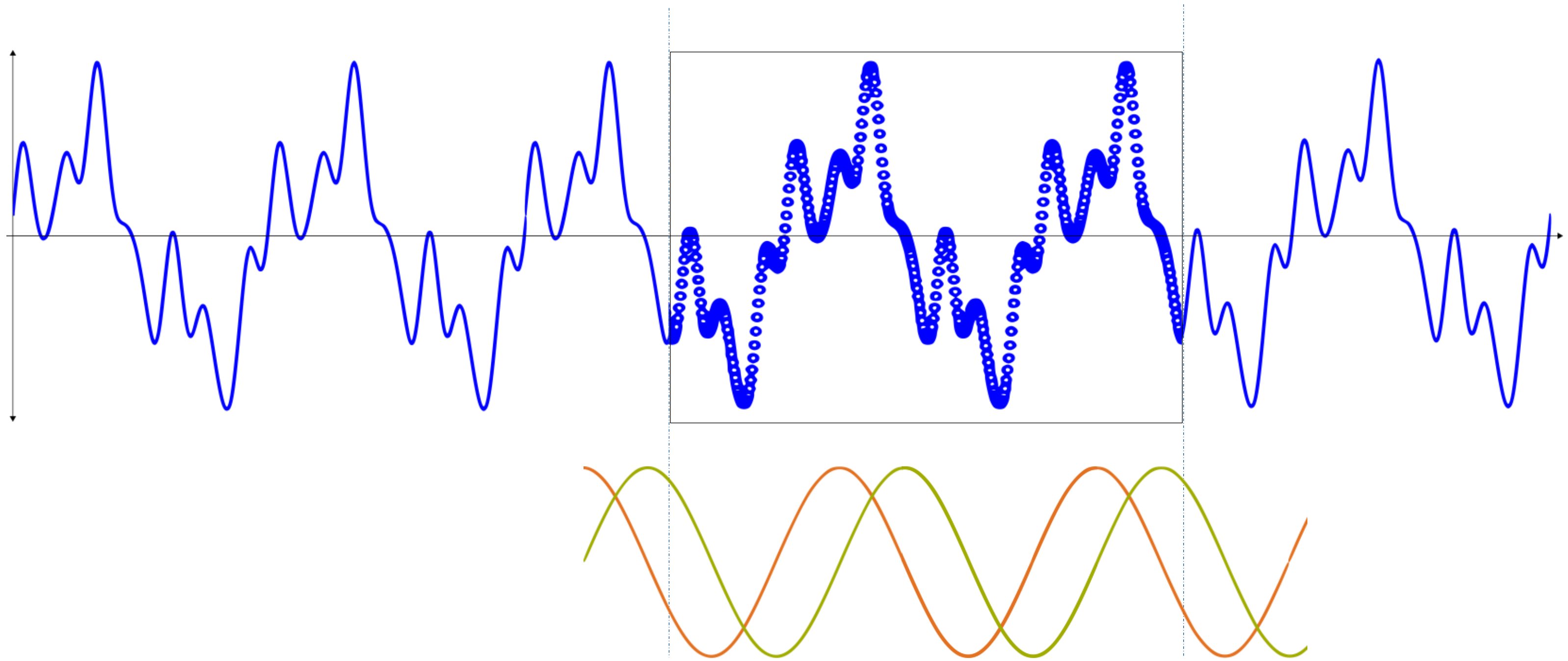
- Discrete fourier transform (DFT)
- Sliding Discrete Fourier Transform (SDFT)
- Modulated Sliding Discrete Fourier Transform (mSDFT)



# Implementation nuances

## Fourier based methods

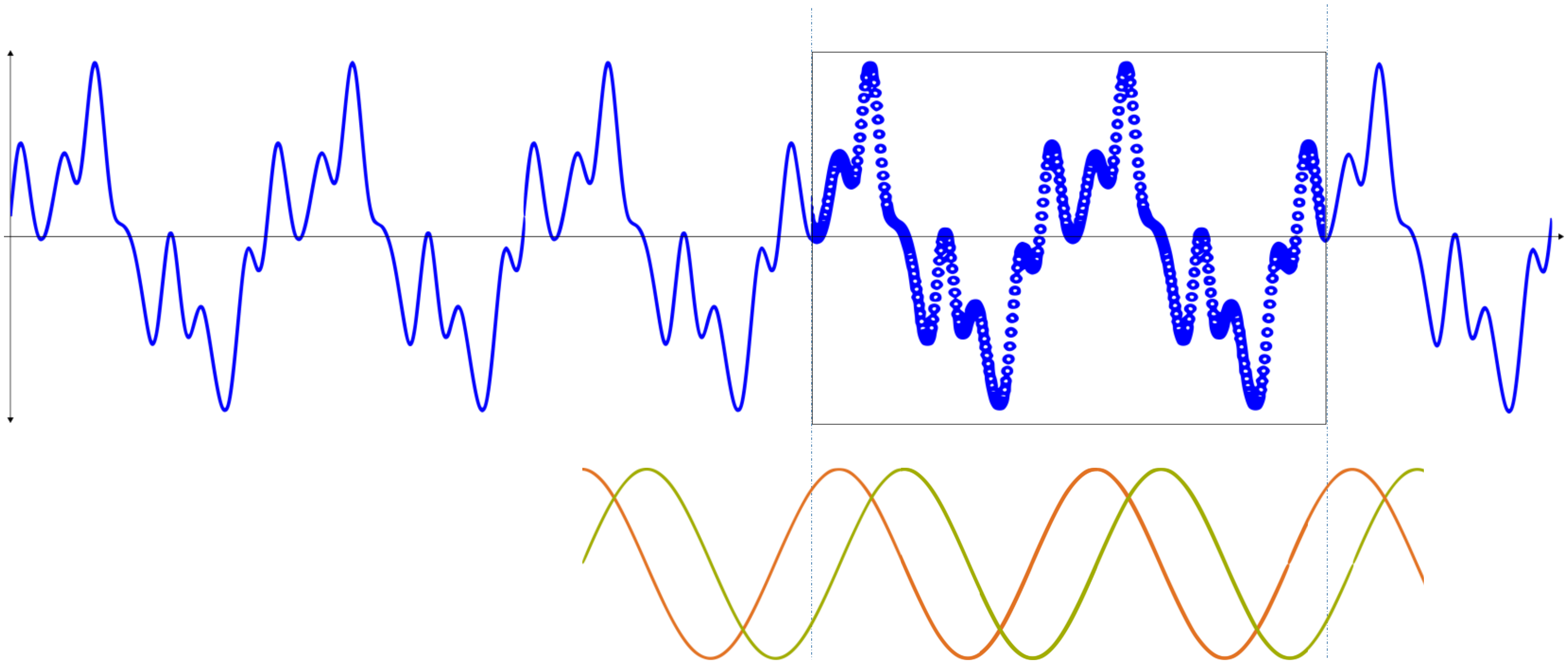
- Discrete fourier transform (DFT)
- Sliding Discrete Fourier Transform (SDFT)
- Modulated Sliding Discrete Fourier Transform (mSDFT)



# Implementation nuances

## Fourier based methods

- Discrete fourier transform (DFT)
- Sliding Discrete Fourier Transform (SDFT)
- Modulated Sliding Discrete Fourier Transform (mSDFT)

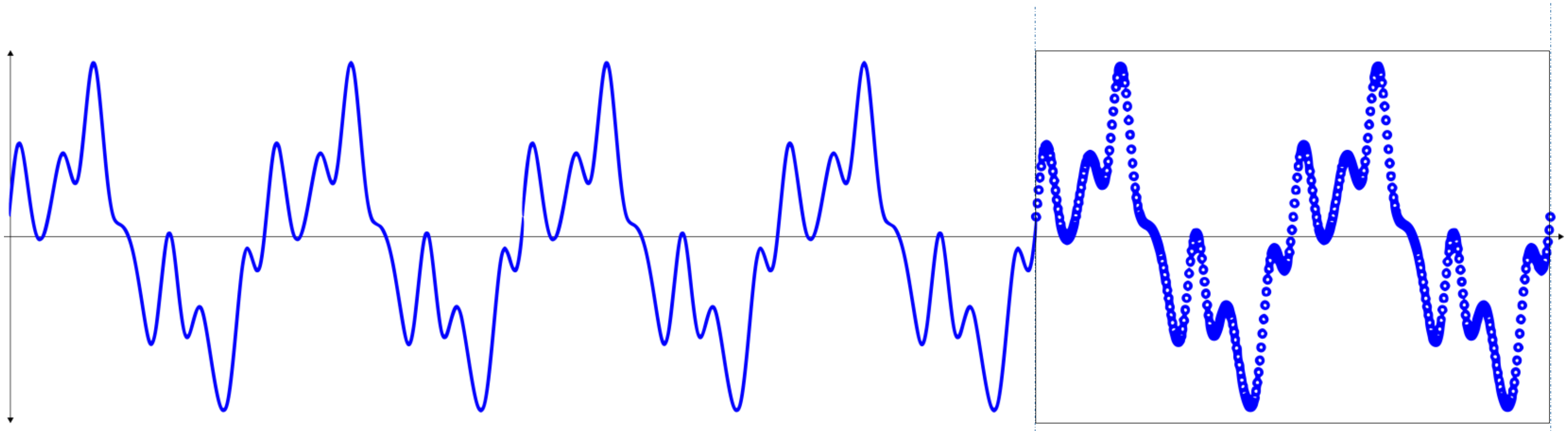


# Implementation nuances

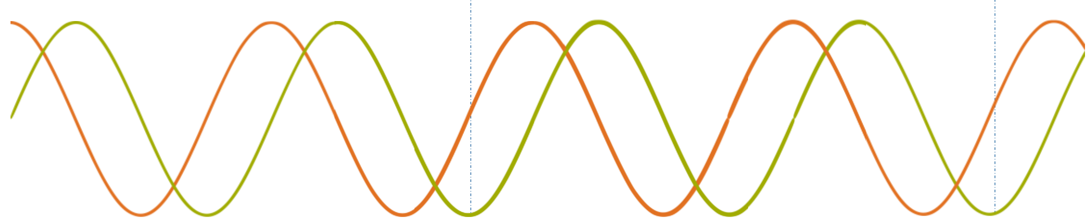
## Fourier based methods

- Discrete fourier transform (DFT)
- Sliding Discrete Fourier Transform (SDFT)
- Modulated Sliding Discrete Fourier Transform (mSDFT)

**Robust against frequency error**



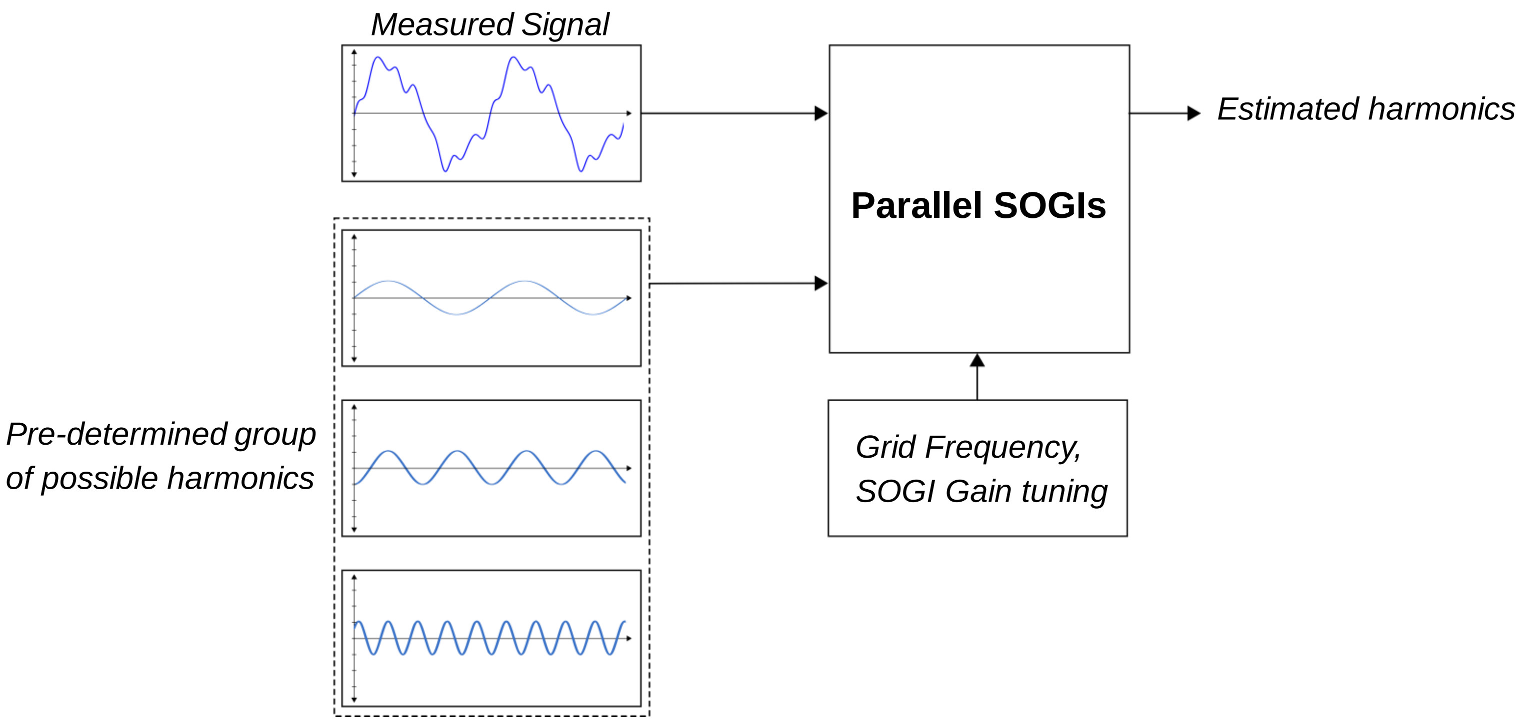
*Grid frequency estimation must be accurate.*



# Implementation nuances

## Time-frequency based methods

- Second Order generalized integrators



# Implementation nuances

## Time-frequency based methods

- Second Order generalized integrators

**Robust against rogue harmonics**