Performance and Applications of Synchronized Waveform Data Compression

Steven Blair
Jason Costello

April 2022

synaptec

Agenda

1. Industry challenges
2. Slipstream: data compression
3. Performance and comparison
4. Applications

synaptec

Industry challenges

Power grids

Reduced system inertia

Increasing levels of IBRs

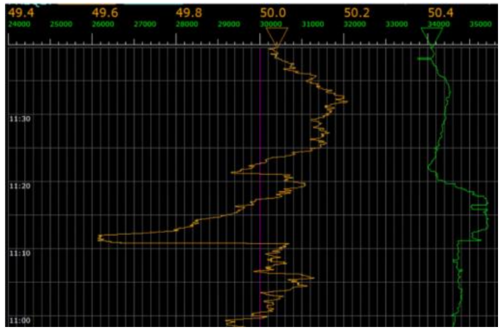
Extremes of weather/wildfires

Monitoring technologies

Phasor-based monitoring has limits

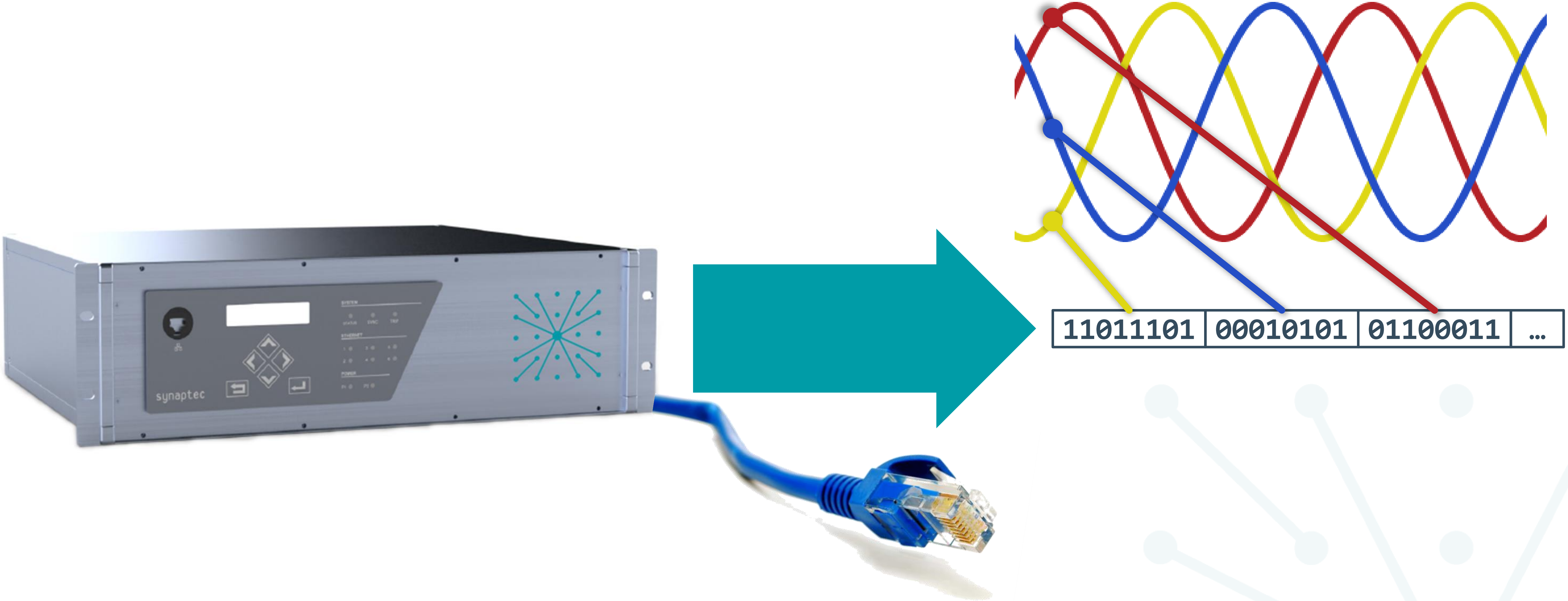
Data comms and analysis can be inefficient

Multiple competing standards

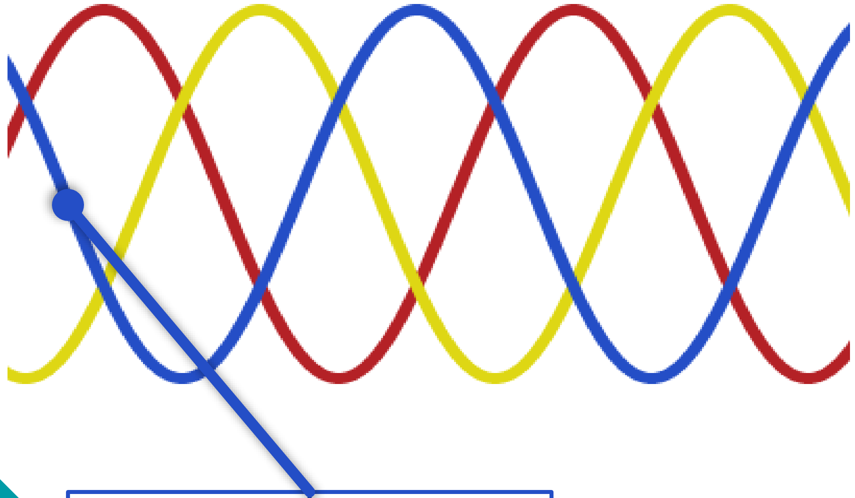


synaptec

Digitising waveform measurements



Digitising waveform measurements



01011101	10010101
01011101	10010101

The data value (eg it’s 40 volts)

01011101	10010101
01011101	10010101
01011101	10010101
01011101	10010101

When was it

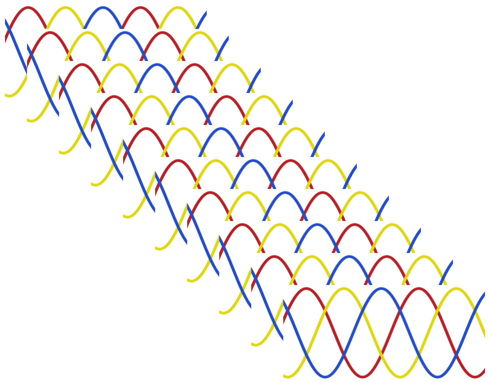
01011101	10010101
01011101	10010101

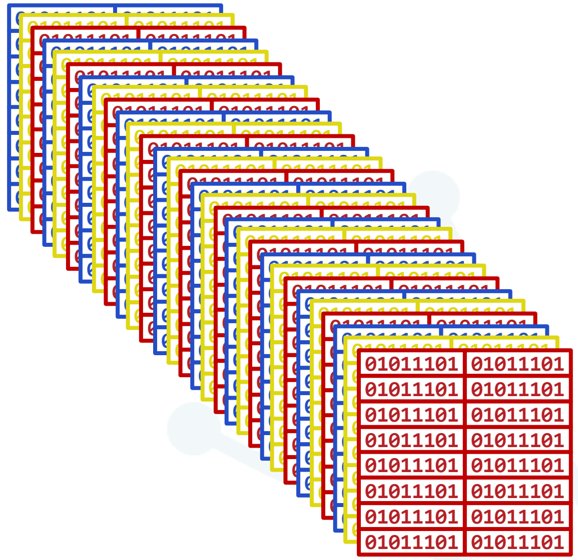
Did something go wrong?

total: 16 bytes

Digitising waveform measurements

up to 30 passive voltage and current sensors per fibre
(equivalent to 5-10 Merging Units)



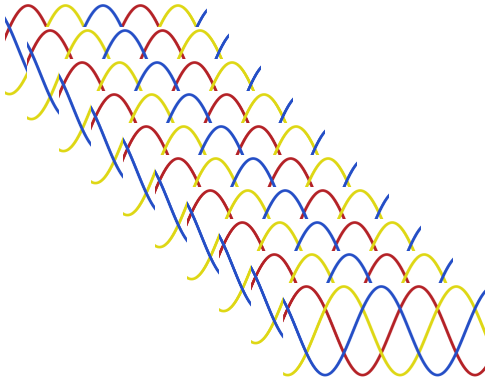


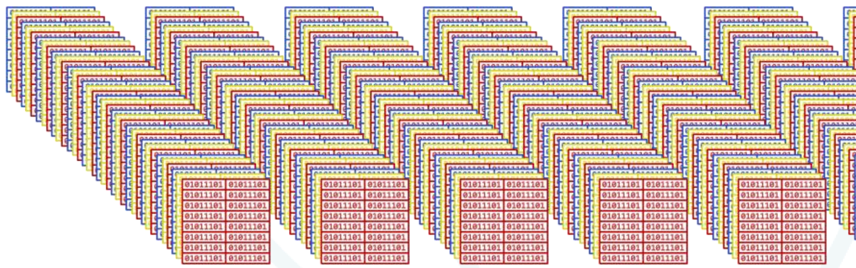
total: 480 bytes,
every sample

synaptec

Digitising waveform measurements

up to 30 passive voltage and current sensors per fibre
(equivalent to 5-10 Merging Units)





do this every 69-250 microseconds

total: 6.9 megabytes
every second
(55.2 Mbps)

synaptec

The data problem

4 kHz sampling:

1.9 megabytes every second → 7 gigabytes every hour → 61 terabytes every year

14.4 kHz sampling:

6.9 megabytes every second → 25 gigabytes every hour → 218 terabytes every year

14.4 kHz is equivalent to:

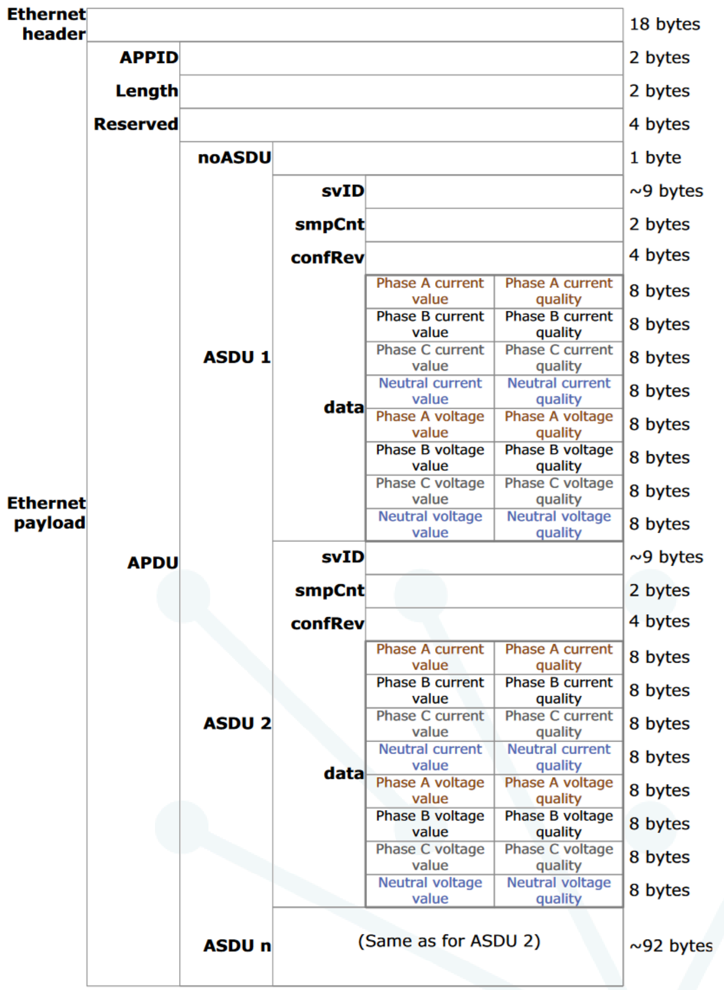
- 4K high-quality video stream
- ~46 Microsoft Teams video streams

synaptec

Synchronized waveform data compression

Objectives:

- Designed for **streaming waveform data**, similar to IEC 61850-9-2 or IEC 61869-9 SV
- Optimised for **smallest message size**
- **Low overhead** compared to SV
- **Lossless**: must not add errors or distortion
- **Flexible**: variable number of samples per message for different applications
- Compress each data stream separately
- Assume out-of-band comms will arrange meta information (like IEEE C37.118.2)

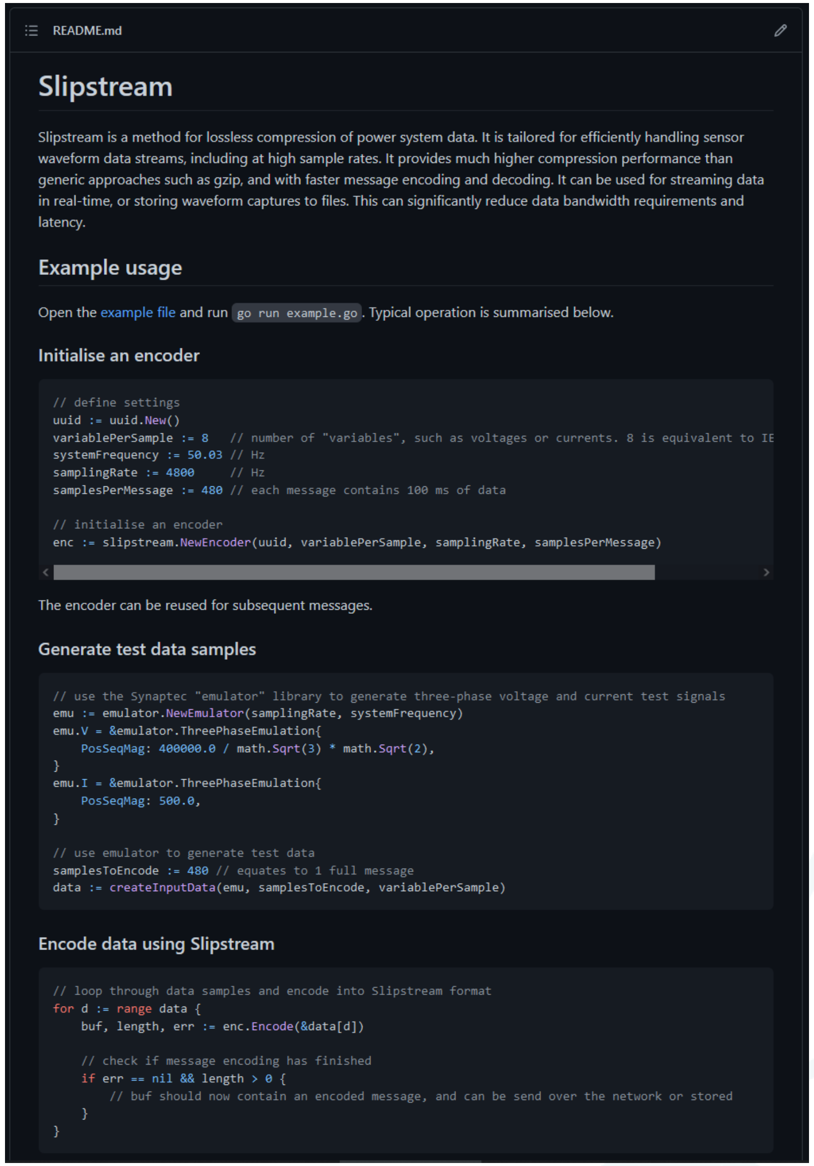


IEC 61850-9-2 SV payload format (~219 bytes for 2 samples)

synaptec

Other compression principles

- Preserve time sync and data quality information
- Produce a byte stream for any transport method (e.g. UDP, TCP, Ethernet, HTTP, WebSocket, raw data file)
- Prefer efficient and fast encoding/decoding methods (simple linear arithmetic)
- Error or loss of a message must not affect other messages
- Open source project: https://github.com/synaptecltd/slipstream



README.md

Slipstream

Slipstream is a method for lossless compression of power system data. It is tailored for efficiently handling sensor waveform data streams, including at high sample rates. It provides much higher compression performance than generic approaches such as gzip, and with faster message encoding and decoding. It can be used for streaming data in real-time, or storing waveform captures to files. This can significantly reduce data bandwidth requirements and latency.

Example usage

Open the example file and run `go run example.go`. Typical operation is summarised below.

Initialise an encoder

```
// define settings
uuid := uuid.New()
variablePerSample := 8   // number of "variables", such as voltages or currents. 8 is equivalent to IE
systemFrequency := 50.03 // Hz
samplingRate := 4800     // Hz
samplesPerMessage := 480 // each message contains 100 ms of data

// initialise an encoder
enc := slipstream.NewEncoder(uuid, variablePerSample, samplingRate, samplesPerMessage)
```

The encoder can be reused for subsequent messages.

Generate test data samples

```
// use the Synaptec "emulator" library to generate three-phase voltage and current test signals
emu := emulator.NewEmulator(samplingRate, systemFrequency)
emu.V = &emulator.ThreePhaseEmulation{
    PosSeqMag: 400000.0 / math.Sqrt(3) * math.Sqrt(2),
}
emu.I = &emulator.ThreePhaseEmulation{
    PosSeqMag: 500.0,
}

// use emulator to generate test data
samplesToEncode := 480 // equates to 1 full message
data := createInputData(emu, samplesToEncode, variablePerSample)
```

Encode data using Slipstream

```
// loop through data samples and encode into Slipstream format
for d := range data {
    buf, length, err := enc.Encode(&data[d])

    // check if message encoding has finished
    if err == nil && length > 0 {
        // buf should now contain an encoded message, and can be send over the network or stored
    }
}
```

Slipstream compression approach

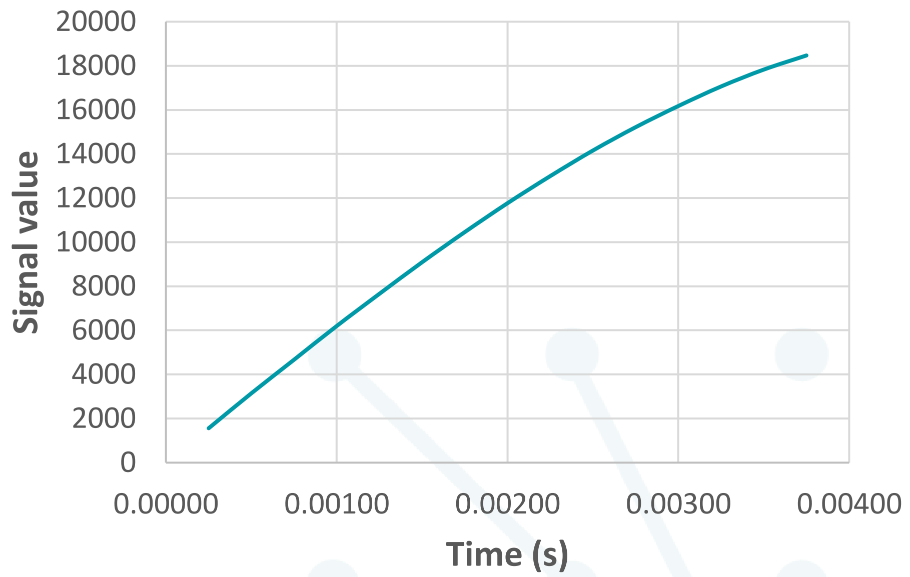
1. delta-delta (multi-layer) with zig-zag encoding
2. simple-8b encoding
3. Optional gzip compression (for large data captures only)
4. Run-length encoding of data quality values

Header	UUID (16 bytes)		
	Starting timestamp (8 bytes)		
	Number of samples encoded (max. 4 bytes)		
First sample	Stream 1 (4 bytes)	Stream 2 (4 bytes)	Stream *n* (4 bytes)
Remaining samples (delta-delta encoded)	Stream 1 delta (~1 byte)	Stream 2 delta (~1 byte)	Stream *n* delta (~1 byte)
	Stream 1 delta (~1 byte)	Stream 2 delta (~1 byte)	Stream *n* delta (~1 byte)
	… (~1 byte)	… (~1 byte)	… (~1 byte)
Quality (RLE)	Stream 1 (>=2 bytes)	Stream 2 (>=2 bytes)	Stream *n* (>=2 bytes)

synaptec

Delta-delta encoding

Time (s)	Signal	delta	delta-delta
0.00025	1554	1554	**1554**
0.00050	3140	1586	**32**
0.00075	4664	1524	**-62**
0.00100	6193	1529	**5**
0.00125	7653	1460	**-69**
0.00150	9086	1433	**-27**
0.00175	10454	1368	**-65**
0.00200	11763	1309	**-59**
0.00225	12991	1228	**-80**
0.00250	14151	1160	**-68**
0.00275	15209	1058	**-103**
0.00300	16169	960	**-98**
0.00325	17054	885	**-74**
0.00350	17821	767	**-119**
0.00375	18462	641	**-126**



Compression performance

Sampling rate (Hz)	Samples per message	Message size (bytes)	Size
4000	10	210	16.4%
4000	4000	40778	8%
14400	6	123	16.9%
14400	14400	2812	0.2%
150000	150000	7238	<0.1%

Slipstream: **0.67 Mbps**
IEC 61850-9-2 SV: **~20 Mbps**

Previous work only achieved ~54% ratio

http://strathprints.strath.ac.uk/57710/1/Blair_etal_AMPS2016_Real_time_compression_of_IEC_61869_9_sampled_value_data.pdf

Assuming a dataset of three-phase voltage and current signals, at 400 kV and 500 A.

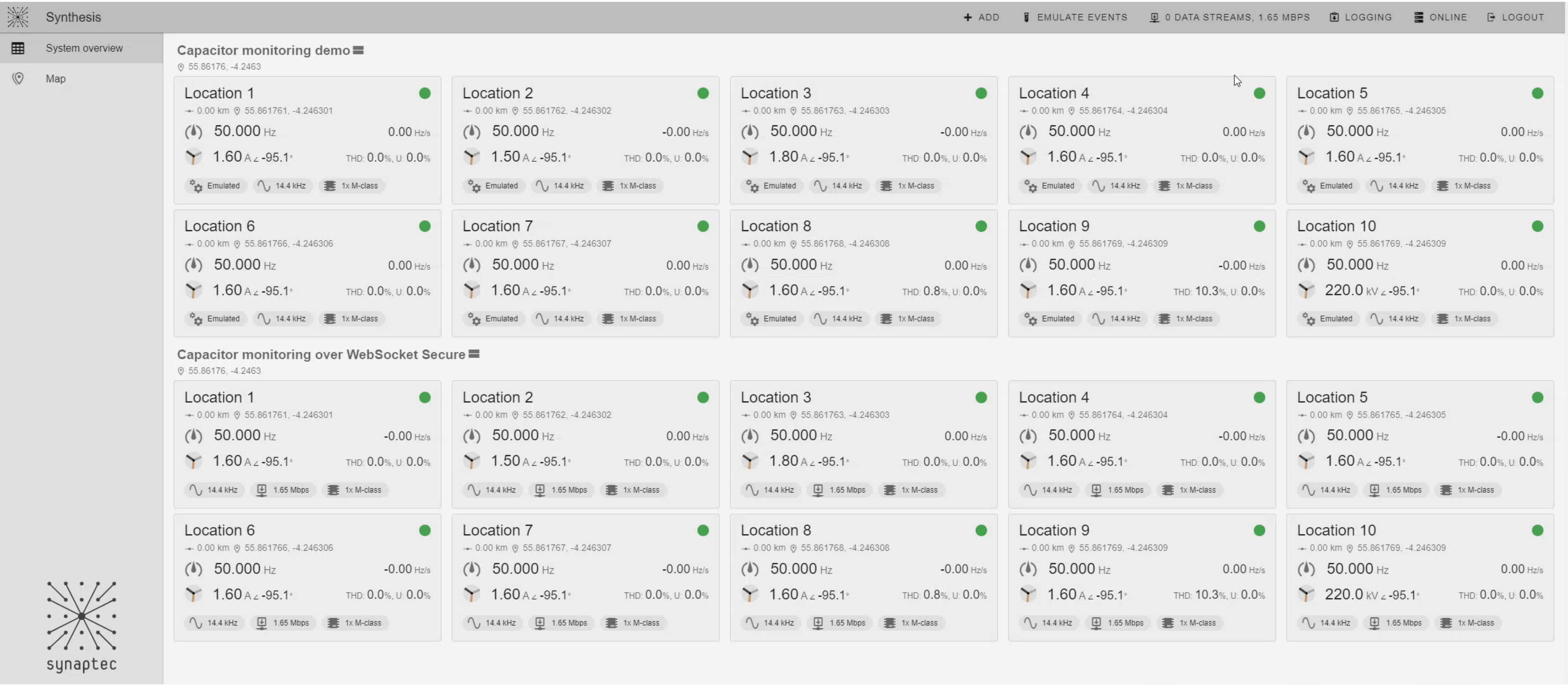
synaptec

Compression performance (with noise and harmonics)

Sampling rate (Hz)	Samples per message	Message size (bytes)	Size
4000	10	236	18.4%
4000	4000	123738	12.1%
14400	6	141	18.3%
14400	14400	123213	6.7%
150000	150000	779918	4.1%

synaptec

Demo: Ten 14.4 kHz streams over WSS in <2 Mbps



Compression is slow?

Data storage method	Sampling rate (Hz)	Samples per message	Message size	Size	Time to encode
Raw data	14400	144000	10.4 MB	56.3%	47 ms
Slipstream	**14400**	**144000**	**0.8 MB**	**4.4%**	**37 ms**
CSV	14400	144000	12.6 MB	68.1%	379 ms
CSV (+gzip)	14400	144000	4.2 MB	22.5%	527 ms

Compressing is faster than not compressing!

Comparison with industry standards

	Slipstream	IEC 61850-9-2 Sampled Values	IEEE C37.118.2	IEEE P2664 (STTP)	PQDIF	COMTRADE
Real-time or storage?	Both	Real-time	Real-time	Real-time	Storage	Storage
Suitable for waveform data?	Yes	Yes	Can be adapted, but not optimal	Yes	Yes	Yes
Payload definition	Fixed datasets	Fixed datasets	Fixed datasets	Dynamic subscription of data points	Arbitrary	Arbitrary
Sampling rate	Any, fixed per dataset	Any, fixed per dataset	Certain reporting rates expected	Any per data point	Any	Any
Open source?	**Yes**	No, but some implementations are	No, but some implementations are	No, but some implementations are	No	No
Security	Handled by transport layer	Auth + encryption possible	Handled by transport layer	Auth + encryption possible	n/a	n/a
Bandwidth/storage use	**Low**	Very high	Medium/high	Low/medium	High	High

synaptec

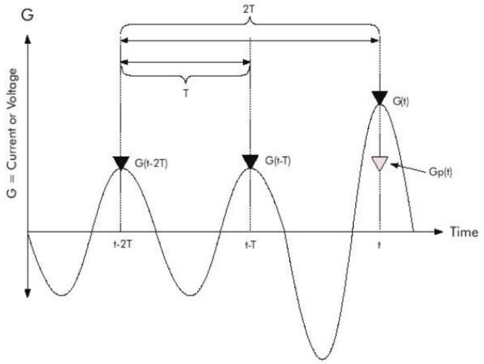
Future improvements

- Optimisation of internal parameters for best compression performance
- Clearly define timestamp and quality fields
- Expand supported data types
- Formalise metadata format and exchange

Applications

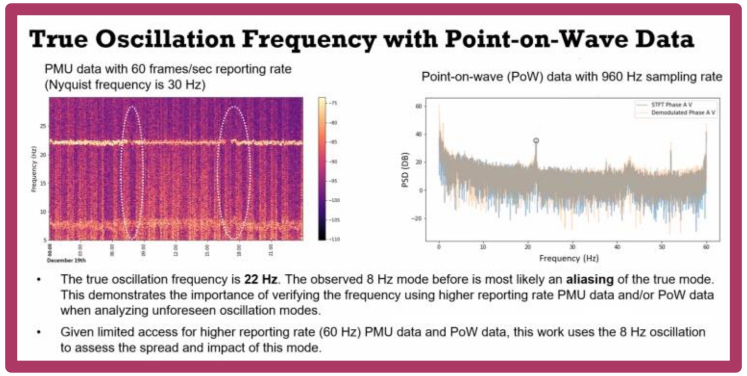
Long event capture and storage

New protection methods based on waveforms, not just phasors



https://doi.org/10.1109/PSCE.2004.1397508

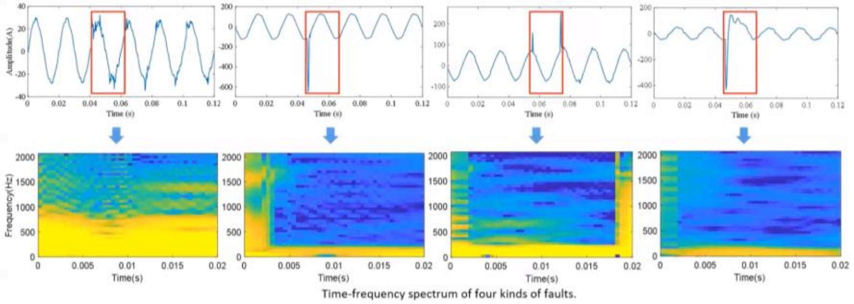
Continuous monitoring of harmonics and other PQ metrics



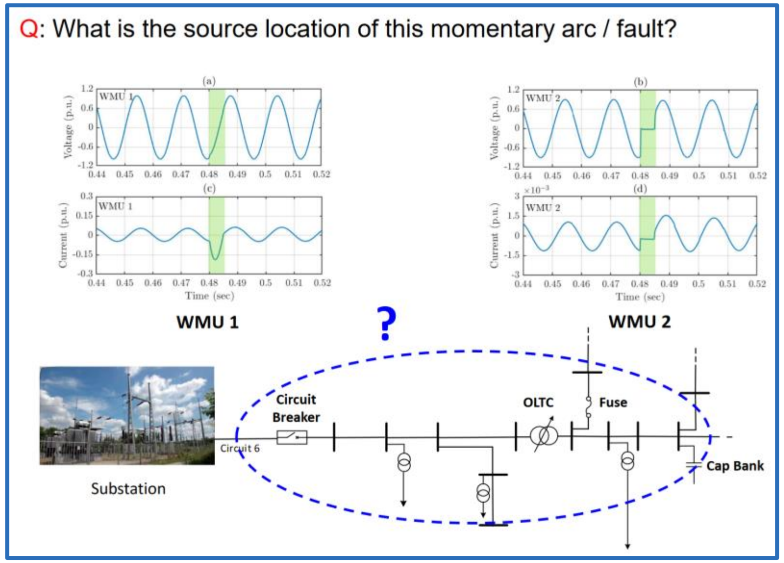
https://www.naspi.org/sites/default/files/2021-04/D1S1_02_wang_dominion_naspi_20210413.pdf

Automated event classification and location

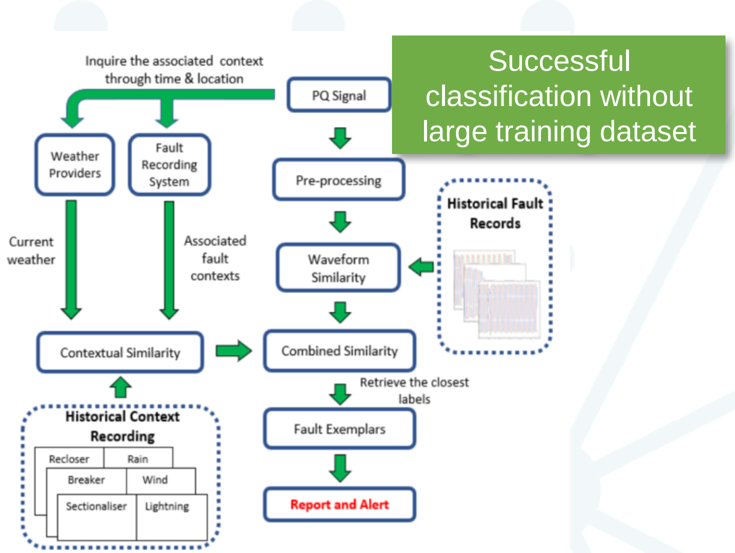
Analysis of transients – condition monitoring of primary assets



Tianshu Bi, NCEPU, IEEE SGSMA 2021



Successful classification without large training dataset



synaptec

M. Izadi and H. Mohsenian-Rad, "Synchronous Waveform Measurements to Locate Transient Events and Incipient Faults in Power Distribution Networks", 2021, doi: 10.1109/TSG.2021.3081017.
Jiang, et al, "Automated Distribution Network Fault Cause Identification with Advanced Similarity Metrics", 2020: https://doi.org/10.1109/TPWRD.2020.2993144

Contact info

Dr Steven Blair
Head of Power Systems Technologies

steven.blair@synapt.ec

https://github.com/synapteclitd/slipstream