

# SDR w laboratorium podstaw telekomunikacji.

Wojciech Sułek

Politechnika Śląska

*wojciech.sulek@polsl.pl*

Tuesday 21<sup>st</sup> November, 2023

# Geneza i plan prezentacji

- „Radio programowalne i kognitywne” – przedmiot na 3 semestrze w nowym planie EiT S2 (15W+30L+15P)
- Sprzęt USRP
- Eksperymenty w GRC: transmisja DSB-SC
- Eksperymenty w GRC: transmisja QPSK
- Dyskusja

Materiały źródłowe wykorzystane w prezentacji pochodzą z:

[www.gnuradio.org](http://www.gnuradio.org)

[www.mathworks.com](http://www.mathworks.com)

B.A. Black, *Introduction to Communication Systems, Lab Based Learning with NI USRP and LabView Communications*, National Instruments 2014.

F.J. Harris, M. Rice, *Multirate Digital Filters for Symbol Timing Synchronization in Software Defined Radios*, IEEE Journal on Selected Areas in Communications, vol. 19, 2001.

Z. Yang, et al., *Design and Implementation of the Digital Costas Loop Based on Software Defined Radio*, IEEE International Conference on Instrumentation, Measurement, Computer, Communication and Control, 2011.

# Sprzęt do dyspozycji – RTL-SDR

DVB-T+FM+DAB 820T2 & SDR: RTL2832U, 25MHz–1,7GHz, do 3,2  
MSPS, tylko odbiornik  
Nieograniczona ilość



# Sprzęt do dyspozycji – National Instruments USRP

USRP-2900/USRP-2901: 70MHz–6GHz, 65KSPS–61MSPS, pasmo do 56MHz, USB 3.0,

USRP-2901: 2x2 MIMO

Aktualnie na stanie: 3 sztuki + 2 sztuki



# Sprzęt do dyspozycji – Analog Devices ADALM-PLUTO

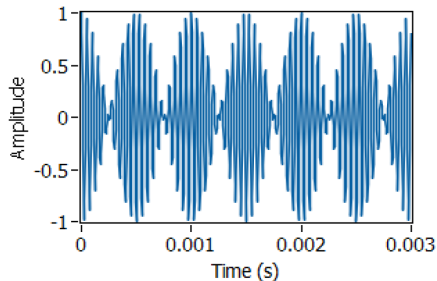
ADALM-PLUTO AD9363, 325MHz–3,8GHz, pasmo do 20MHz, USB 2.0,  
maks. ok. 4-5MSPS

Aktualnie na stanie: 8 sztuk



# Transmisja DSB-SC

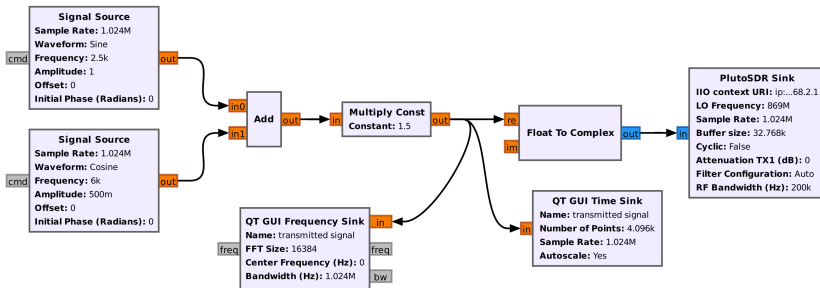
$$g(t) = Am(t) \cos(2\pi f_c t)$$



Modulator: w części "analogowej" urządzenia peryferyjnego SDR.

# Transmisja DSB-SC – nadajnik

## Schemat blokowy w GRC (GNU Radio Companion)



W odbiorniku: wymnożenie przez lokalny oscylator  $\rightarrow$  sygnał w pasmie podstawowym  $r(t)$ .

Przy błędzie fazy nośnej  $\theta = \theta(t)$ , sygnał ten, na wejściu części cyfrowej (programowej) radia można opisać:

$$r_I(t) = \frac{D}{2}m(t) \cos(\theta)$$

$$r_Q(t) = \frac{D}{2}m(t) \sin(\theta)$$

lub też:

$$\tilde{r}(t) = \frac{D}{2}m(t) \cos(\theta) + j\frac{D}{2}m(t) \sin(\theta) = \frac{D}{2}m(t)e^{j\theta}$$



$$\tilde{r}(nT_s) = \frac{D}{2}m(nT_s) \cos(\theta) + j\frac{D}{2}m(nT_s) \sin(\theta) = \frac{D}{2}m(nT_s)e^{j\theta}$$

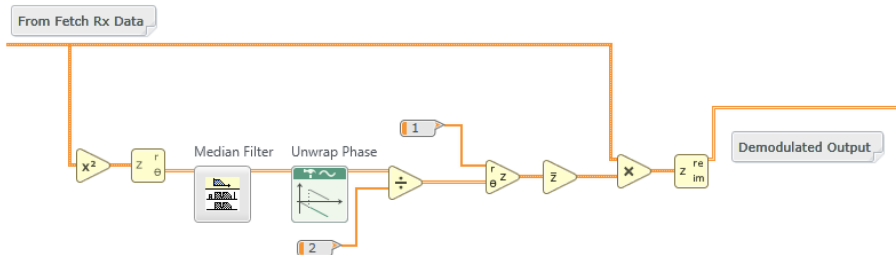
Procedura korekcji błędu  $\theta(nT_s)$ :

- Dla każdej próbki sygnału  $\tilde{r}(nT_s)$  określić (estymować) wartość  $\theta(nT_s)$
- Dokonać korekcji:  $\tilde{r}(nT_s)e^{-j\theta} = \frac{D}{2}m(nT_s)e^{j\theta}e^{-j\theta} = \frac{D}{2}m(nT_s)e^{j0}$

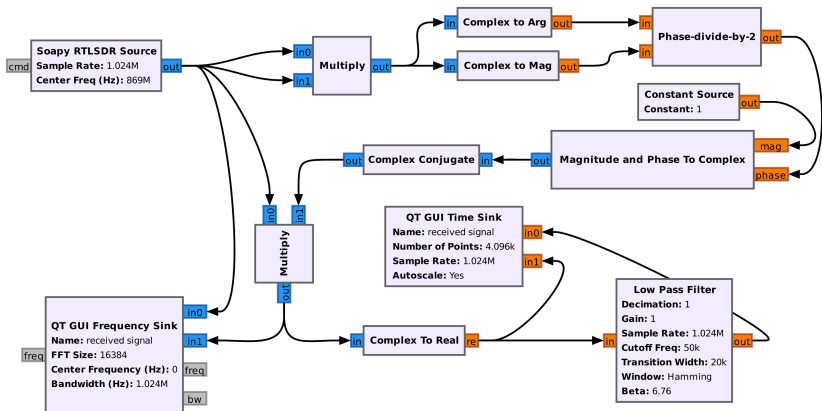
Estymacja  $\theta$  (dla każdej próbki sygnału):

- Wyznaczyć  $\tilde{r}^2(nT_s) = \frac{D^2}{4}m^2(nT_s)e^{j2\theta}$
- Wyznaczyć argument i podzielić przez 2:  $\arg(\tilde{r}^2(nT_s))/2 = \theta$ , a następnie "rozwinąć fazę" (usunąć przesłoki  $\pm\pi$ )

# Transmisja DSB-SC – odbiornik z instrukcji National Instruments (LabView)



# Transmisja DSB-SC – odbiornik



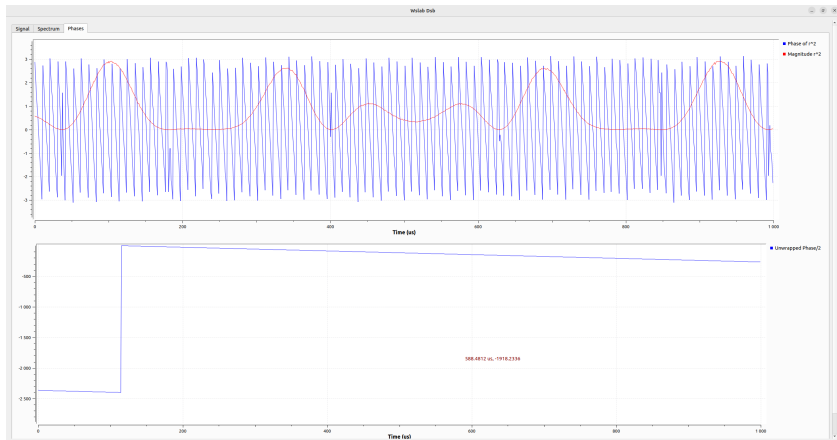
# Transmisja DSB-SC – rozwijanie fazy – blok Phase-divide-by-2

```
1 from gnuradio import gr
2 import numpy as np
3 # ... ..
4 class blk(gr.sync_block):
5     def __init__(self):
6         gr.sync_block.__init__(
7             self,
8             name='Phase-divide-by-2',
9             in_sig=[np.float32, np.float32],
10            out_sig=[np.float32]
11        )
12        self.state = 0
13        self.lastPhaseHop = 0
14        self.previous_sample = 0;
15        self.lastPhaseHopInd = 0
16        self.lastPhaseHop = 0
17 # ... ..
```

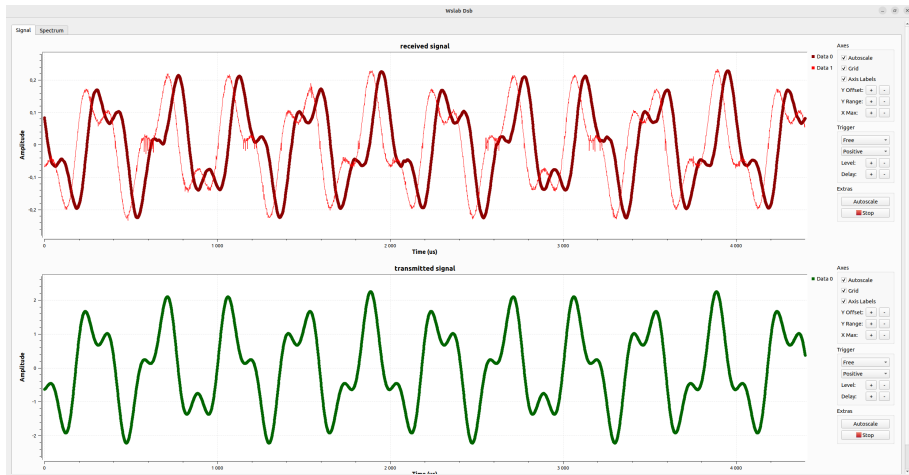
# Transmisja DSB-SC – rozwijanie fazy – blok Phase-divide-by-2

```
1 # ... ..
2 def work(self, input_items, output_items):
3     phase = input_items[0]
4     deltaph = np.diff(phase, prepend=self.previous_sample)
5     cumulativeState = self.state+np.cumsum(deltaph<-np.pi)
6     -np.cumsum(deltaph>np.pi)
7     phaseShiftValue = 2*np.pi*( cumulativeState )
8     output_items[0][:] = (phase+phaseShiftValue)/2
9     self.state = np.mod(cumulativeState[-1], 2)
0     self.previous_sample = phase[-1]
1     return len(output_items[0])
```

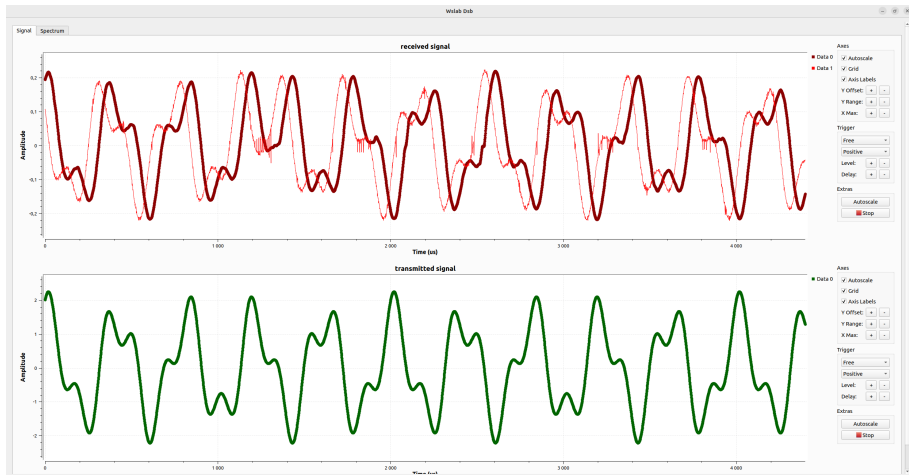
# Transmisja DSB-SC – rozwinięcie fazy – blok Phase-divide-by-2



# Transmisja DSB-SC – sygnał informacyjny w nadajniku PLUTO i odebrany, zdemodulowany w RTL

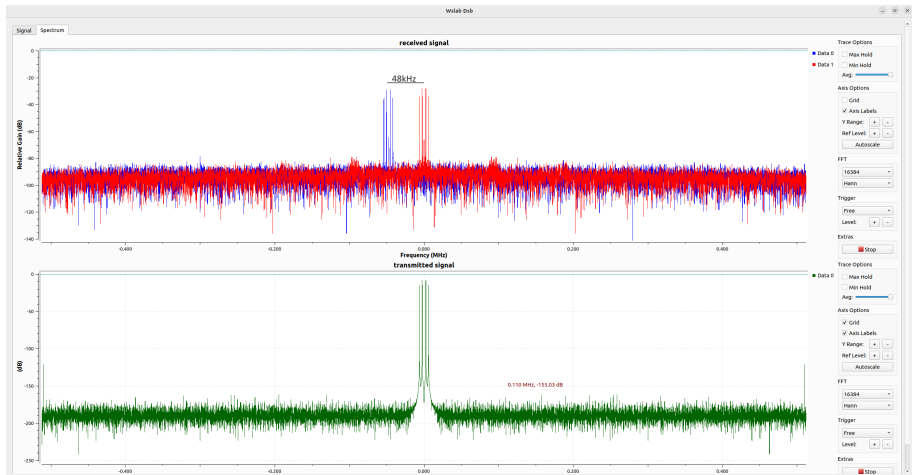


# Transmisja DSB-SC – sygnał informacyjny w nadajniku PLUTO i odebrany, zdemodulowany w RTL

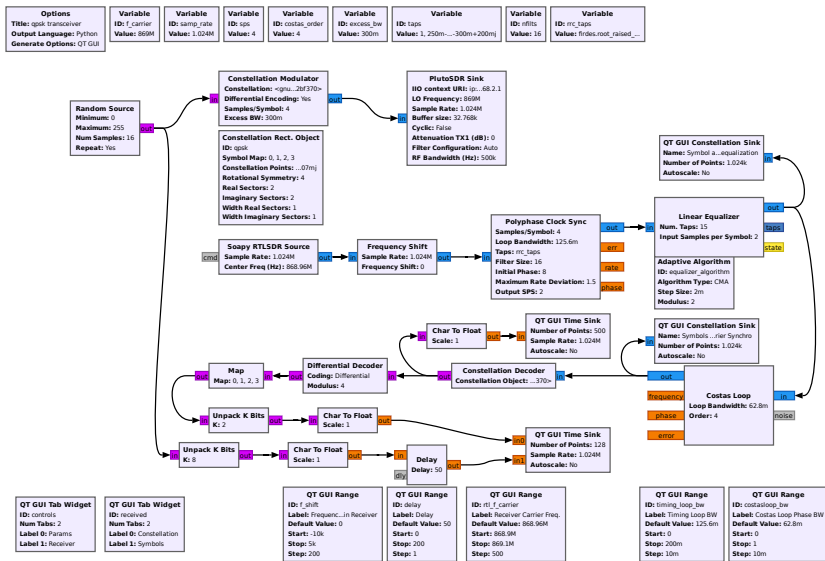




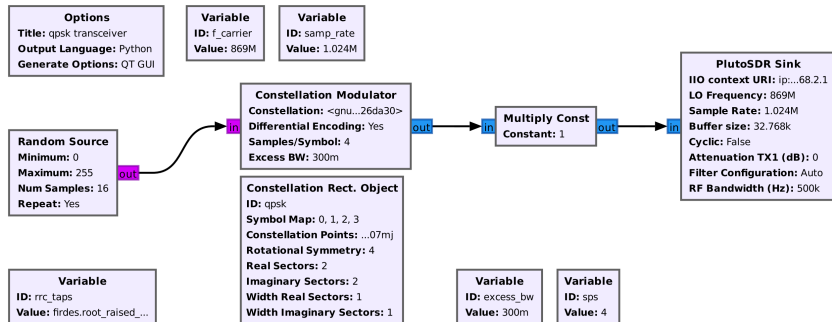
# Transmisja DSB-SC – widma: sygnał inf. w nadajniku, sygnał w odbiorniku przed i po korekcji błędu lokalnego generatora nośnej



# QPSK – nadajnik i odbiornik, 869MHz, 256 kb/s



# QPSK – nadajnik, 869MHz, 256 kb/s



# QPSK – nadajnik, 869MHz, 256 kb/s

The screenshot shows the GNU Radio interface with a 'Constellation Rect. Object' properties dialog box open. The dialog has four tabs: 'General', 'Advanced', 'Documentation', and 'Generated Code'. The 'General' tab is selected, showing the following parameters:

Parameter	Value
ID	qpsk
Symbol Map	[0, 1, 2, 3]
Constellation Points	[0.707+0.707j, -0.707+0.707j, -0.707-0.707j, 0.707-0.707j]
Rotational Symmetry	4
Real Sectors	2
Imaginary Sectors	2
Width Real Sectors	1
Width Imaginary Sectors	1
Soft bits precision	8

In the background, the 'Constellation Modulator' block is visible with the following settings:

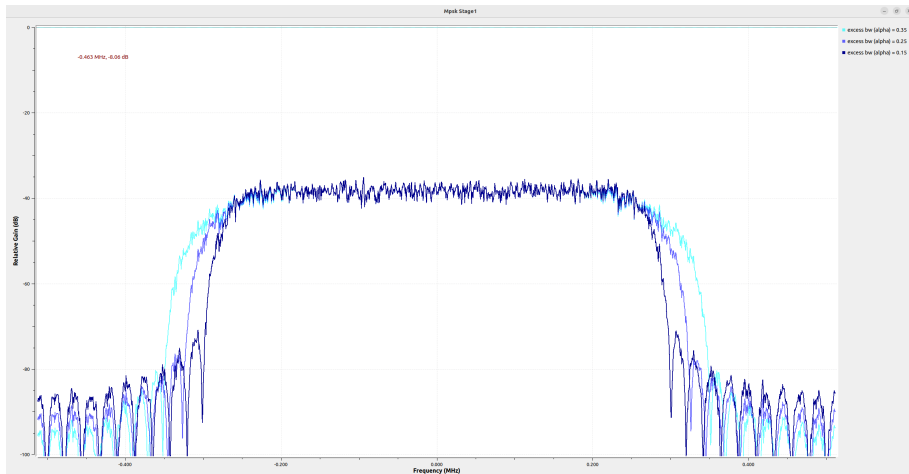
- Constellation: <gnu...d6beb0>
- Differential Encoding: Yes
- Samples/Symbol: 4
- Excess BW: 300m

Below it, the 'Soapy RTLSDR Source' block is visible with the following settings:

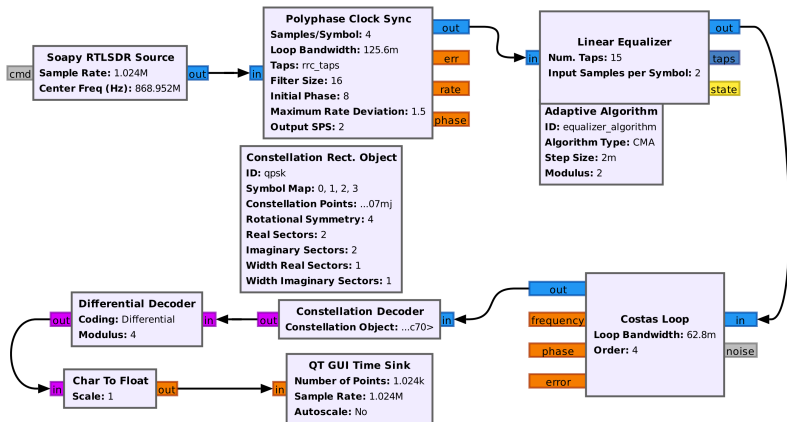
- Sample Rate: 1.024M
- Center Freq (Hz): 868.952M

# QPSK – nadajnik – widmo w pasmie podstawowym

Różne współczynniki poszerzenia pasma (excess bandwidth,  $\alpha$ ) filtra kształtującego RRC:  $\alpha = 0.35$ ,  $\alpha = 0.25$ ,  $\alpha = 0.15$

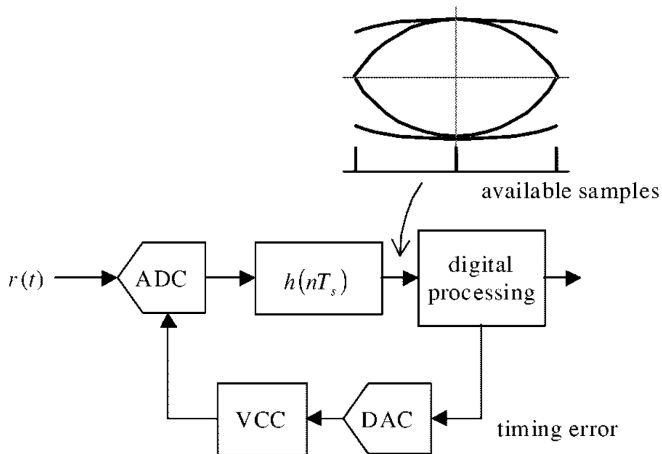


# QPSK – odbiornik, 869MHz, 256 kb/s

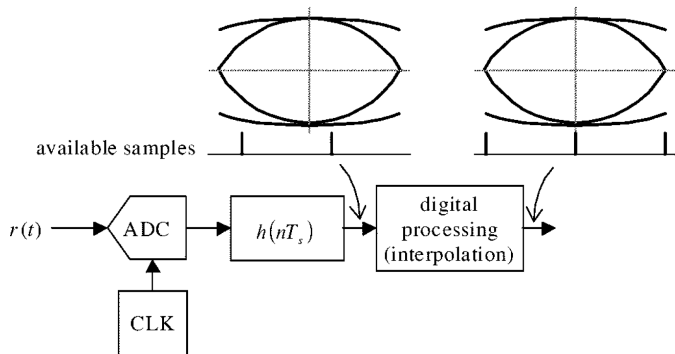


# Clock Sync – synchronizacja symbolowa

F.J. Harris, M. Rice, *Multirate Digital Filters for Symbol Timing Synchronization in Software Defined Radios*, IEEE Journal on Selected Areas in Communications, vol. 19, 2001.

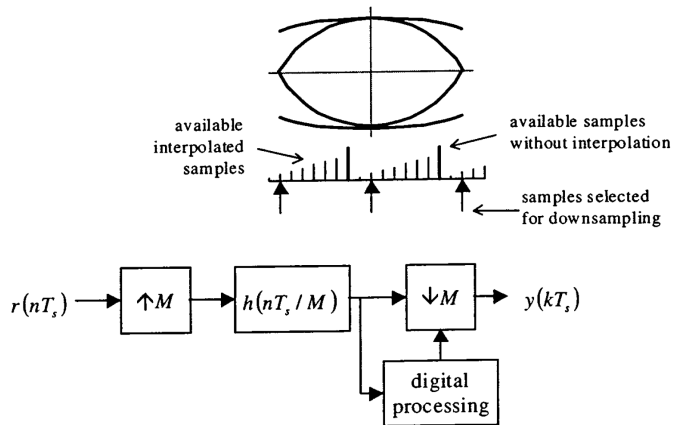


# Clock Sync – synchronizacja symbolowa

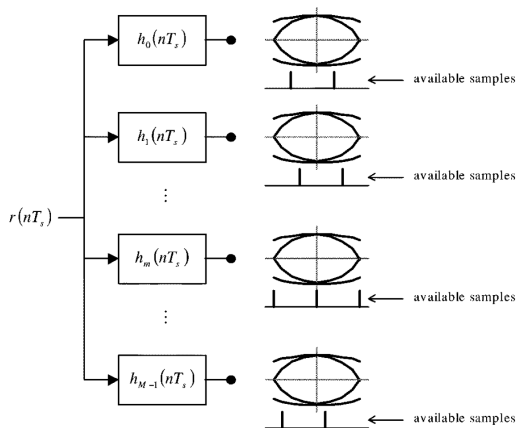




# Clock Sync – synchronizacja symbolowa

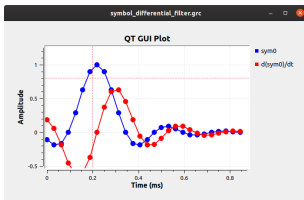
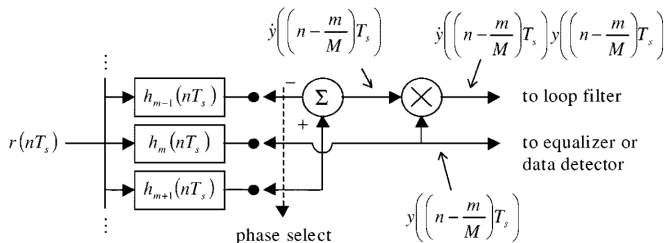


# Clock Sync – synchronizacja symbolowa – bank filtrów wielofazowych (polyphase filter bank)

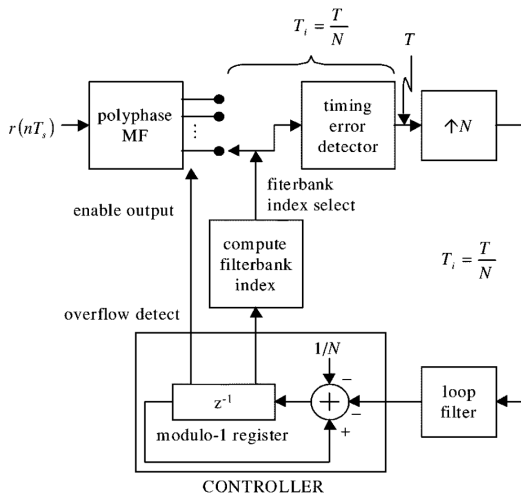


$$h_m(nT_s) = h\left(nT_s + \frac{m}{M}T_s\right)$$

# Clock Sync – synchronizacja symbolowa – bank filtrów: detekcja błędu fazy



# Clock Sync – synchronizacja symbolowa – bank filtrów wielofazowych (polyphase filter bank)



# Linear Equalizer – ślepa korekcja charakterystyki kanału

Algorytm CMA (Constant Modulus Algorithm)

Filtr sprowadzający moduł odbieranych symboli do wartości jednostkowej.

Minimalizacja:

$$\text{Mean} (|y[m]|^2 - 1)^2$$

Współczynniki filtra  $q_l[m]$ ,  $l = -L_q, \dots, L_q$  zmieniane w kroku  $m$  w przeciwną stronę do gradientu:

$$q_l[m + 1] = q_l[m] - \frac{d}{dq_l[m]} (|y[m]|^2 - 1)^2$$

co można sprowadzić do ( $\mu$  – step size):

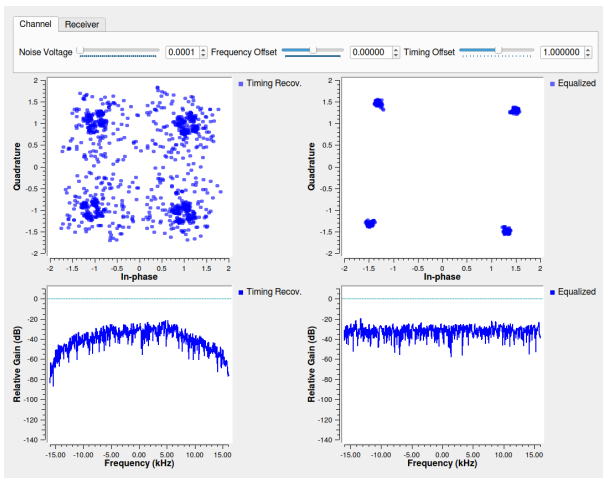
$$q_l[m + 1] = q_l[m] - \mu \cdot y^*[m] \cdot z[m - l] \cdot (|y[m]|^2 - 1)$$

dla każdego  $l = -L_q, \dots, L_q$

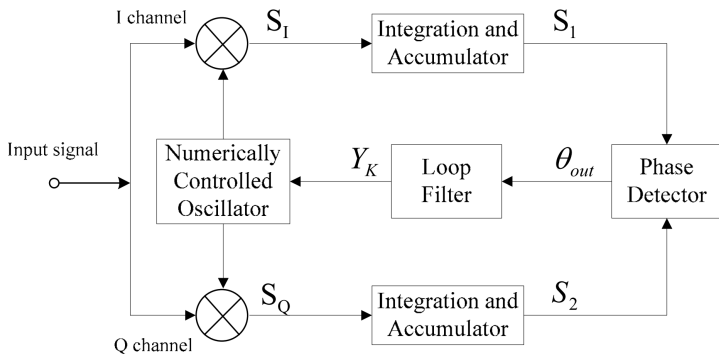
gdzie  $z[m]$  – wejście korektora,  $y[m]$  – wyjście korektora

# Linear Equalizer – ślepa korekcja charakterystyki kanału

Efekt korekcji CMA (symulacja – kanał 4-drogowy, przy idealnej synchronizacji symbolowej i synchronizacji nośnej)



# Costas Loop – korekcja błędów częstotliwości i fazy (phase and frequency offset)

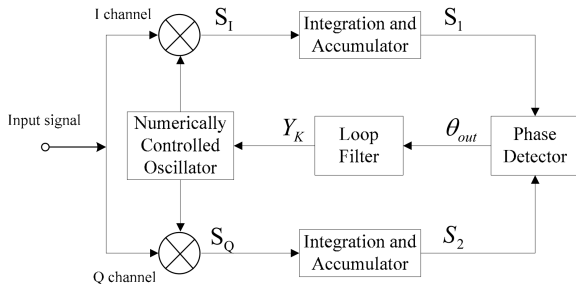


Wejście: sygnał pośredniej częstotliwości (IF). Generator NCO generuje:

$$I(t) = \cos(\omega_0 t + \theta)$$

$$Q(t) = \sin(\omega_0 t + \theta)$$

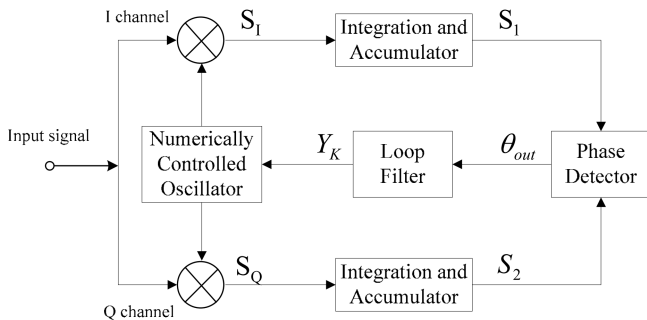
# Costas Loop – korekcja błędów częstotliwości i fazy (phase and frequency offset)



$$\begin{aligned} S_I(t) &= m(t) \cos w_0 t \times \cos (w_0 t + \theta) \\ &= \frac{1}{2} m(t) [\cos \theta + \cos (2w_0 t + \theta)] \\ S_Q(t) &= m(t) \cos w_0 t \times \sin (w_0 t + \theta) \\ &= \frac{1}{2} m(t) [\sin \theta + \sin (2w_0 t + \theta)] \end{aligned}$$



# Costas Loop – korekcja błędów częstotliwości i fazy (phase and frequency offset)

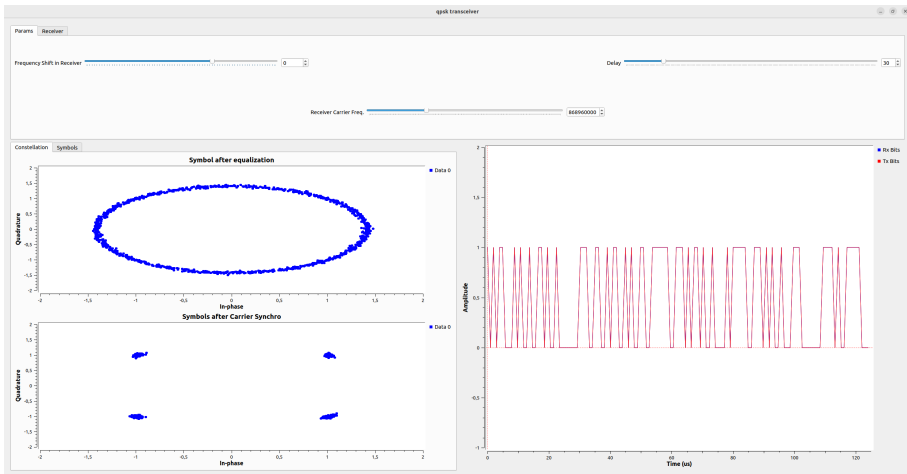


$$S_1(t) = \frac{1}{2}m(t) \cos \theta$$

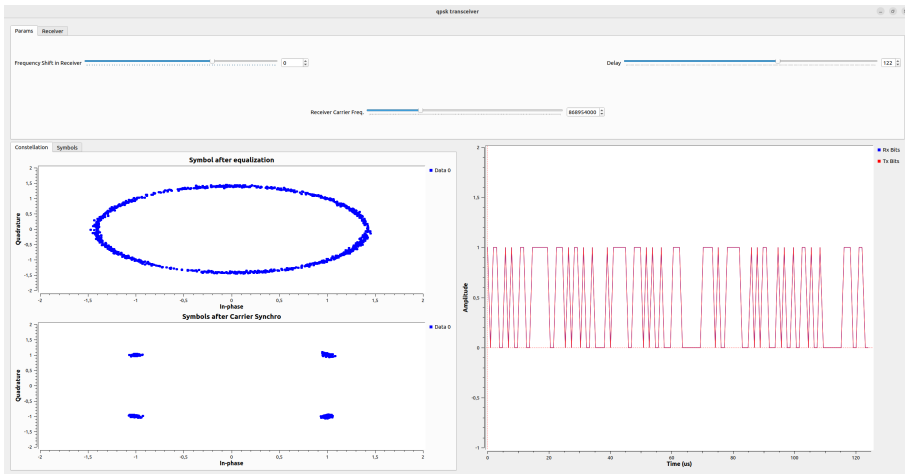
$$S_2(t) = \frac{1}{2}m(t) \sin \theta$$

$$\theta_{out}(t) = \arctg(S_2(t)/S_1(t)) = \theta$$

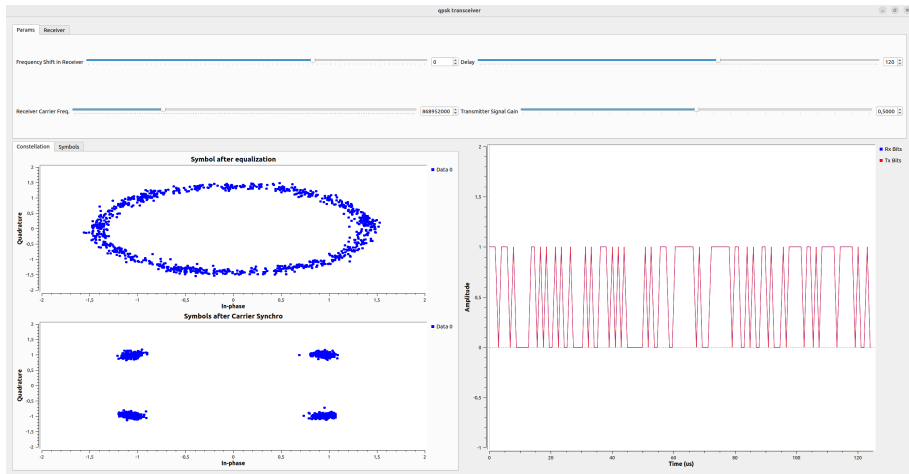
# QPSK – odbiornik, moduł RTL



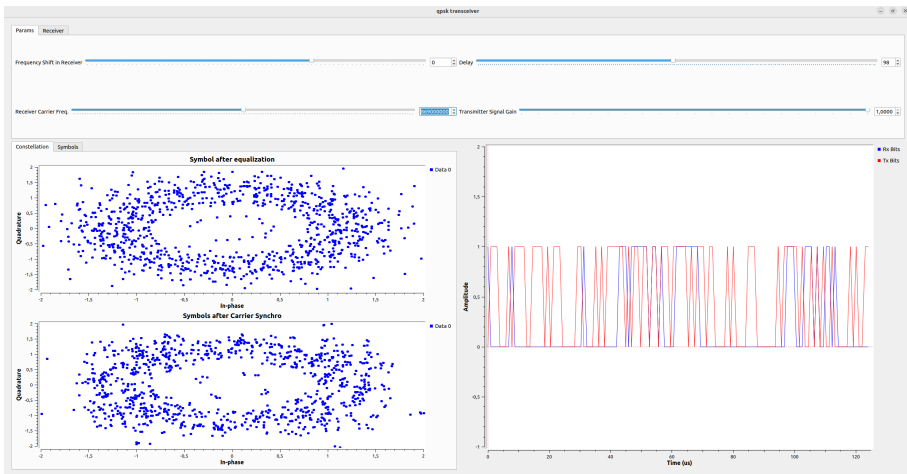
# QPSK – odbiornik, moduł RTL



# QPSK – odbiornik, moduł RTL

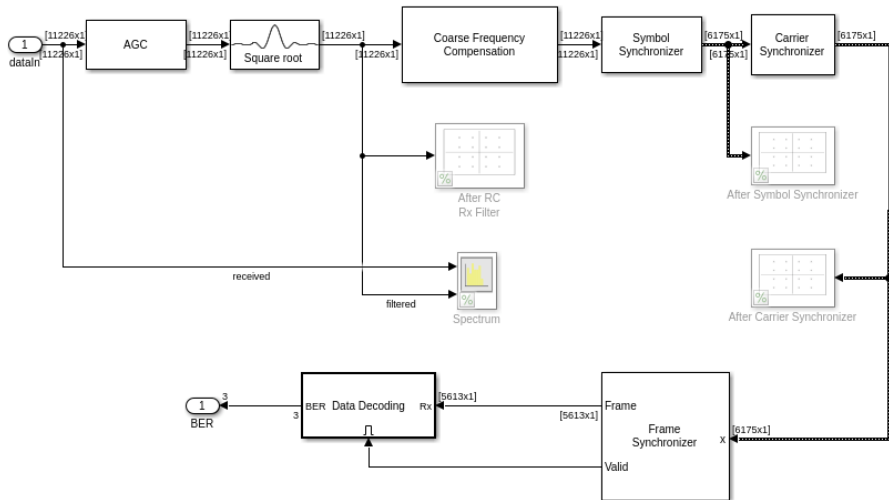


# QPSK – odbiornik, moduł RTL



# QPSK – Matlab/Simulink – odbiornik w SDR

## Receiver



# Zagadnienia do dyskusji

- Narzędzia: Python / GNU Radio Companion / Matlab+Simulink / C++ ?
- Lista tematów laboratoriów / rozdziałów

# Spis tematów ćwiczeń / rozdziałów książki – instrukcje

## National Instruments

- Audio source, Audio sink, File source, File sink, etc.
- Introduction to the USRP
- Amplitude Modulation
- Frequency-Division Multiplexing
- Image Rejection
- Double-Sideband Suppressed-Carrier
- Frequency Modulation
- Amplitude-Shift Keying
- Frequency-Shift Keying
- Binary Phase-Shift Keying (BPSK)
- The Eye Diagram
- Equalization
- Quadrature Phase-Shift Keying