Implementacja i walidacja układów dyskretnych

dr hab. inż. Dominik Sierociuk

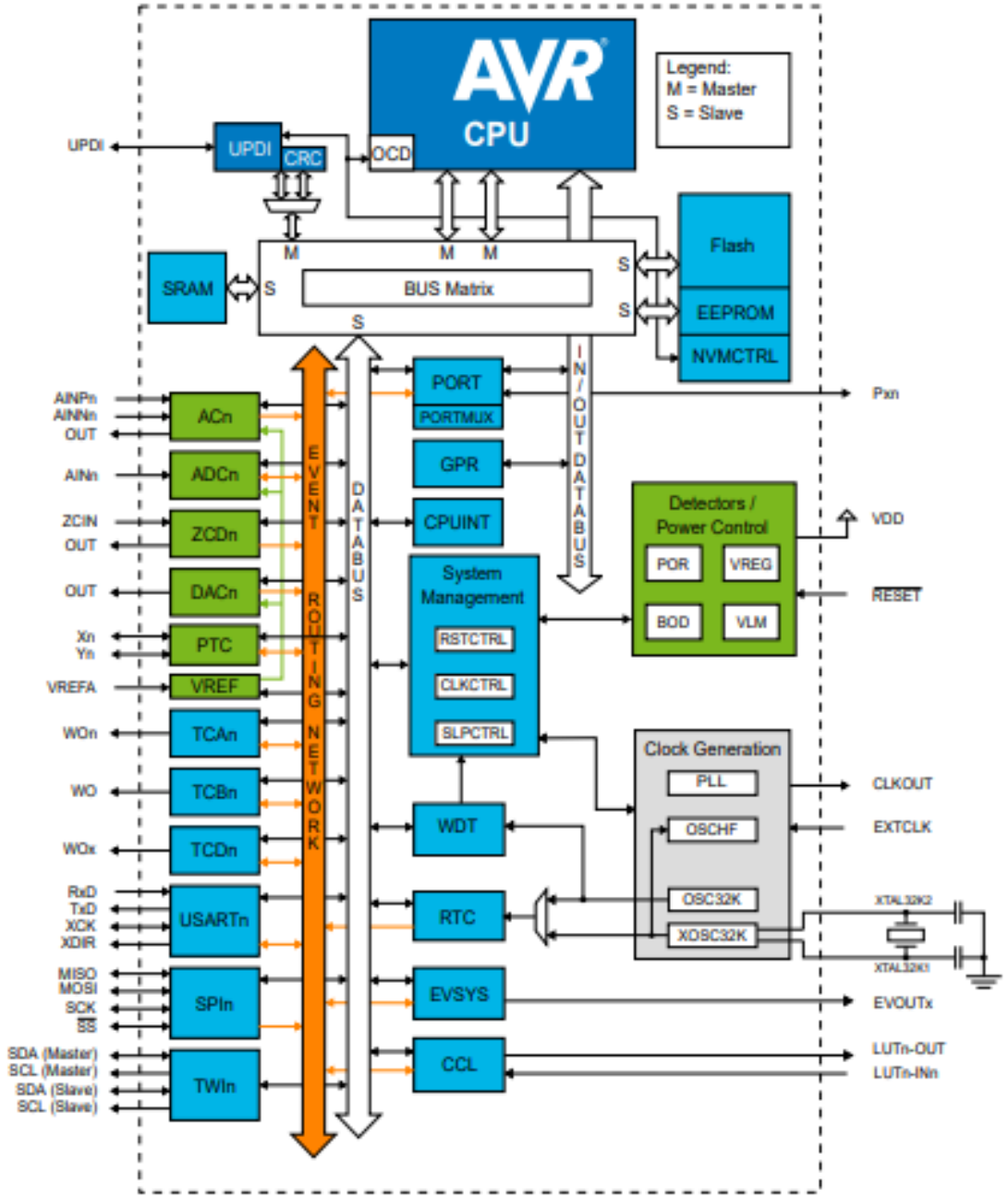
Materiały zostały współfinansowane przez Unię Europejską ze środków Europejskiego Funduszu Społecznego w ramach Programu Operacyjnego Wiedza Edukacja Rozwój 2014-2020, projekt „NERW PW Nauka-Edukacja-Rozwój-Współpraca”

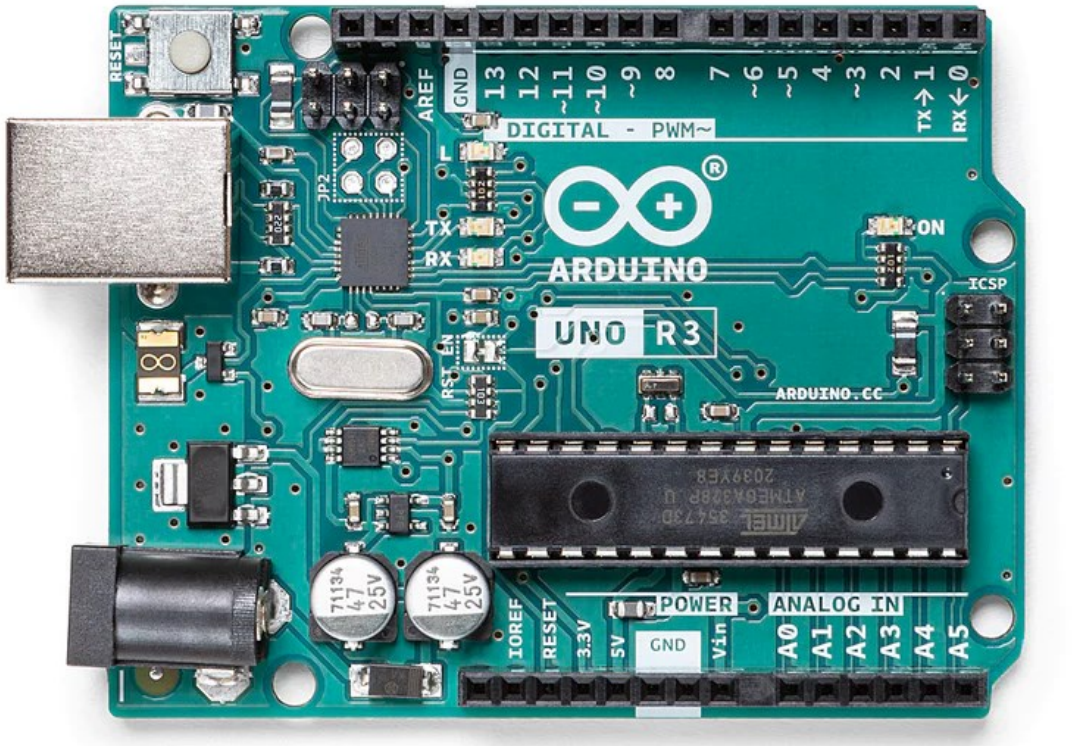
Fundusze Europejskie
Wiedza Edukacja Rozwój

Rzeczpospolita Polska

Unia Europejska
Europejski Fundusz Społeczny

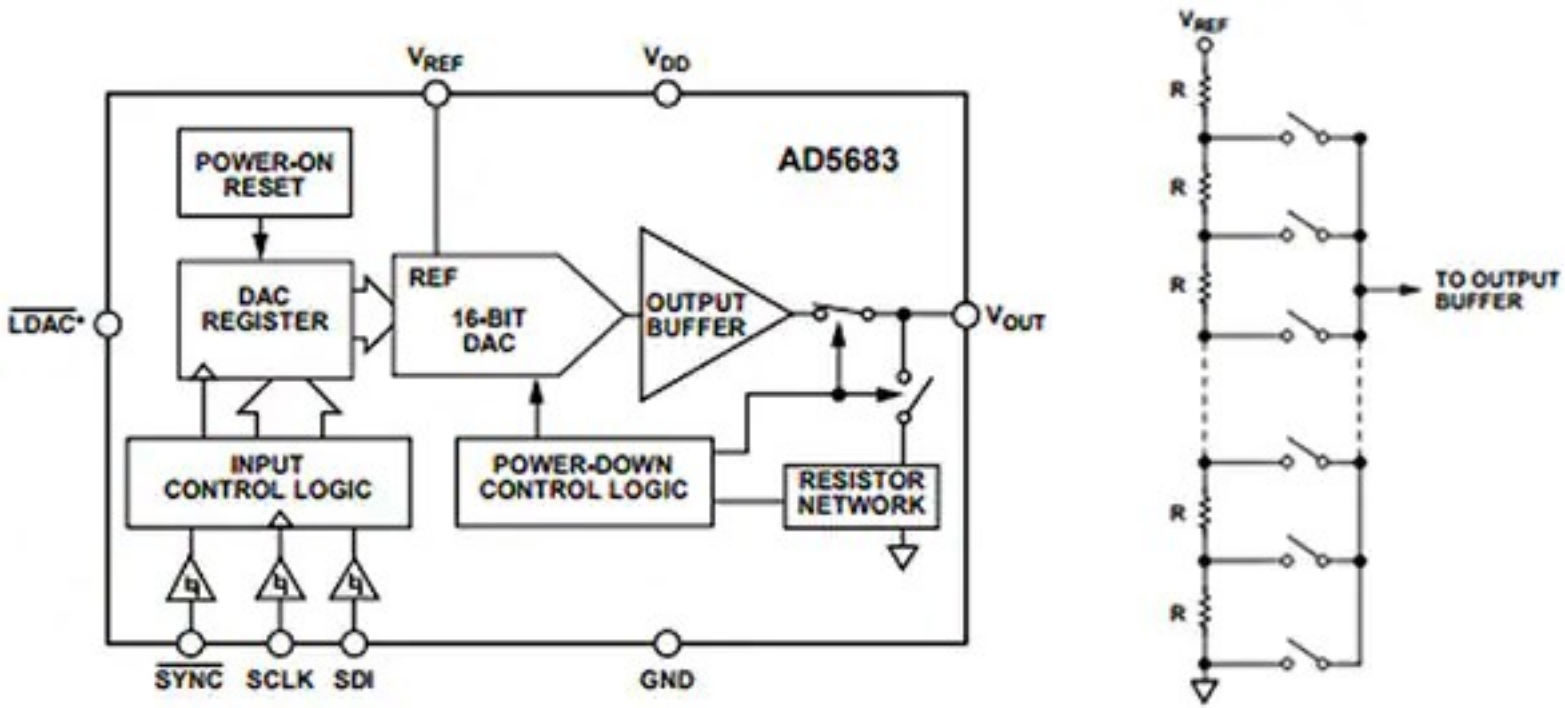
Mikrokontrolery





https://store.arduino.cc/products/arduino-uno-rev3

Przetwornik DAC (cyfrowo-analogowy)



https://www.digikey.pl/pl/articles/adc-dac-tutorial

Przetwornik ADC (analogowo-cyfrowy)

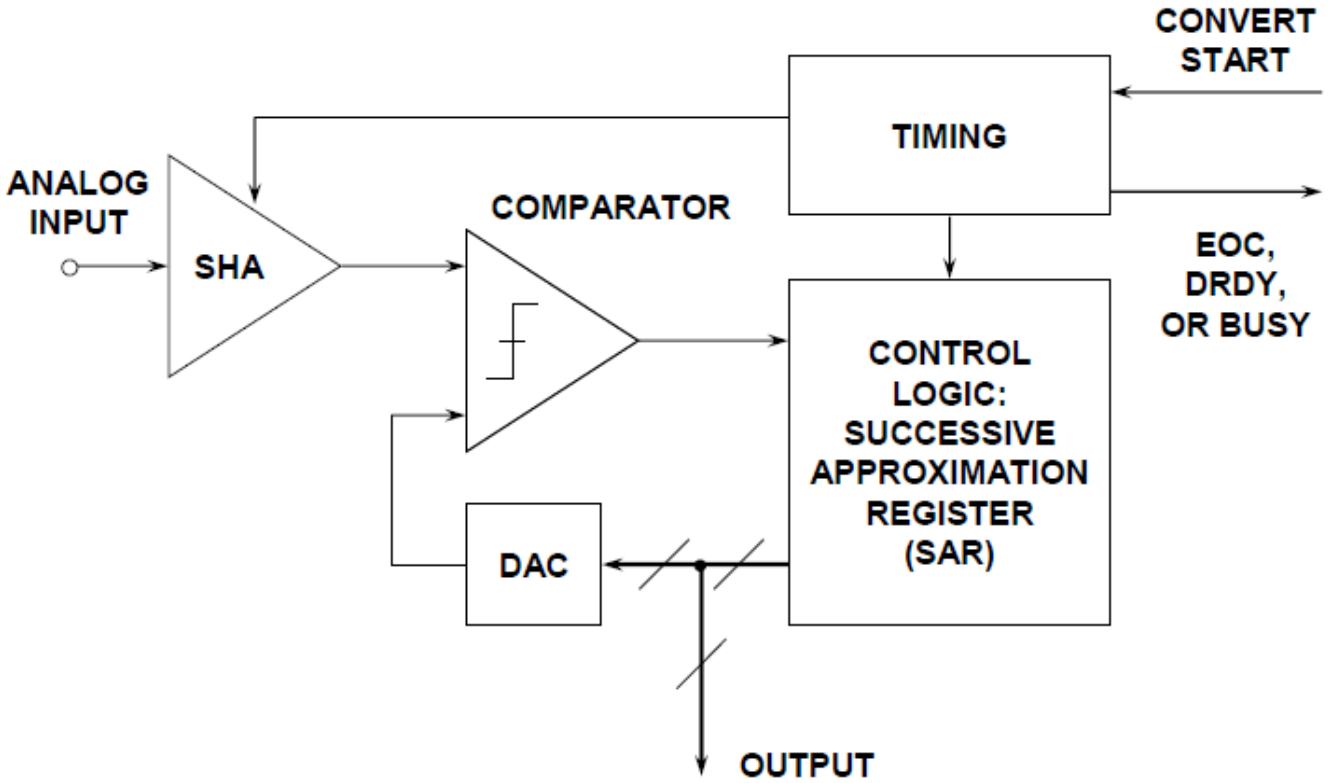
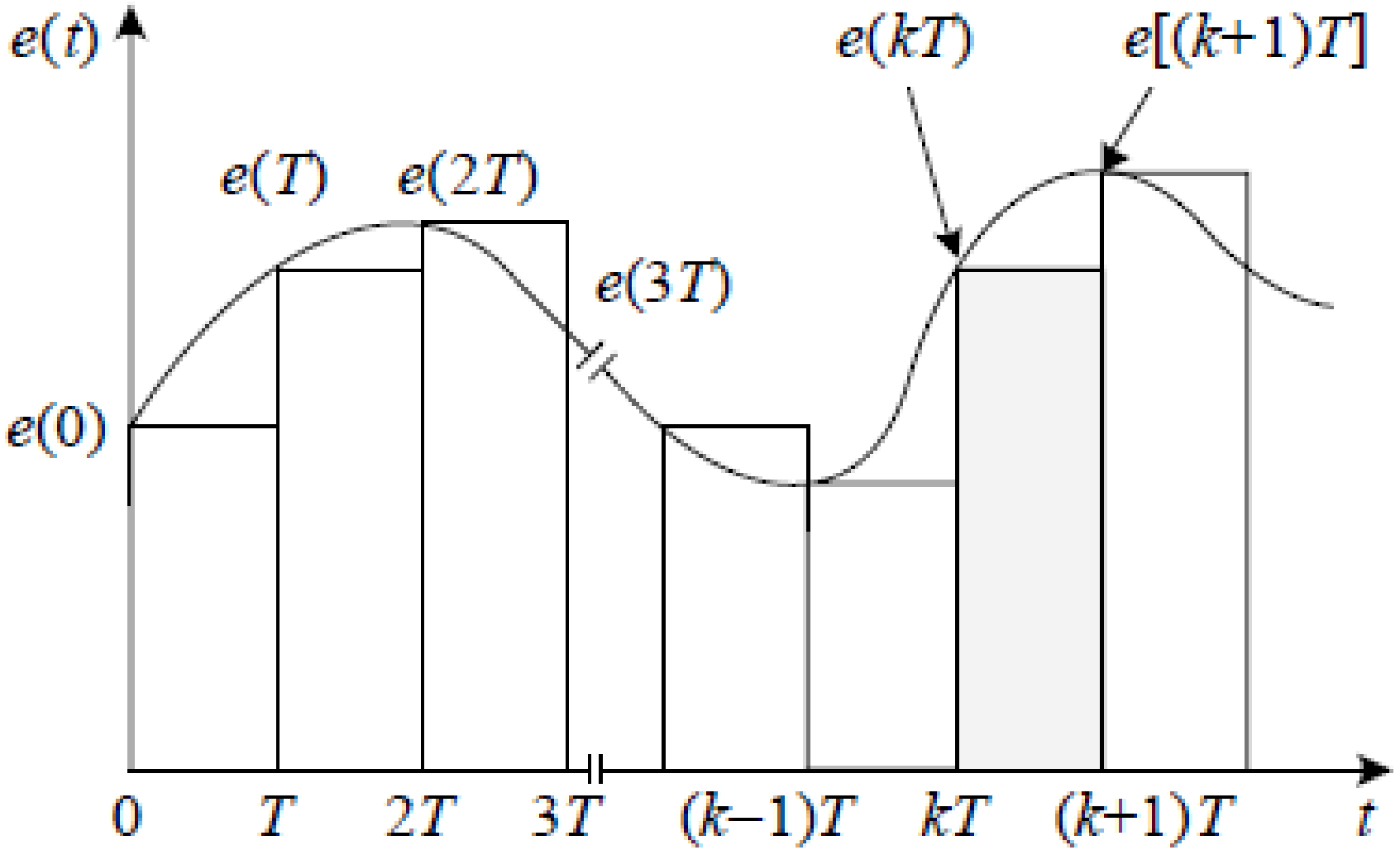


Figure 1: Basic Successive Approximation ADC (Feedback Subtraction ADC)

https://www.digikey.pl/pl/pdf/a/analog-devices/adc-architectures-ii-successive-approximation-adcs

Sygnały dyskretne



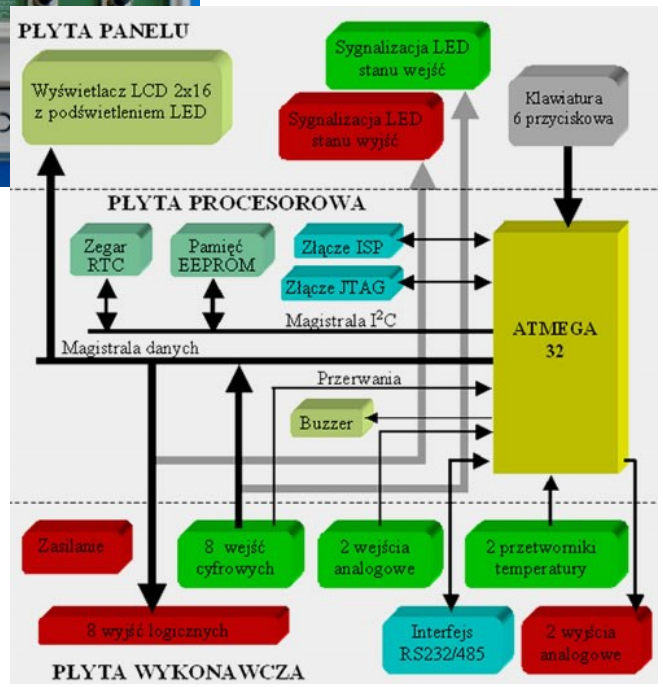
$u_k, u_{k-1}, \dots, u_1, u_0$

$y_k, y_{k-1}, \dots, y_1, y_0$

Transformata Z

$$G(z) = \frac{Y(z)}{U(z)}$$

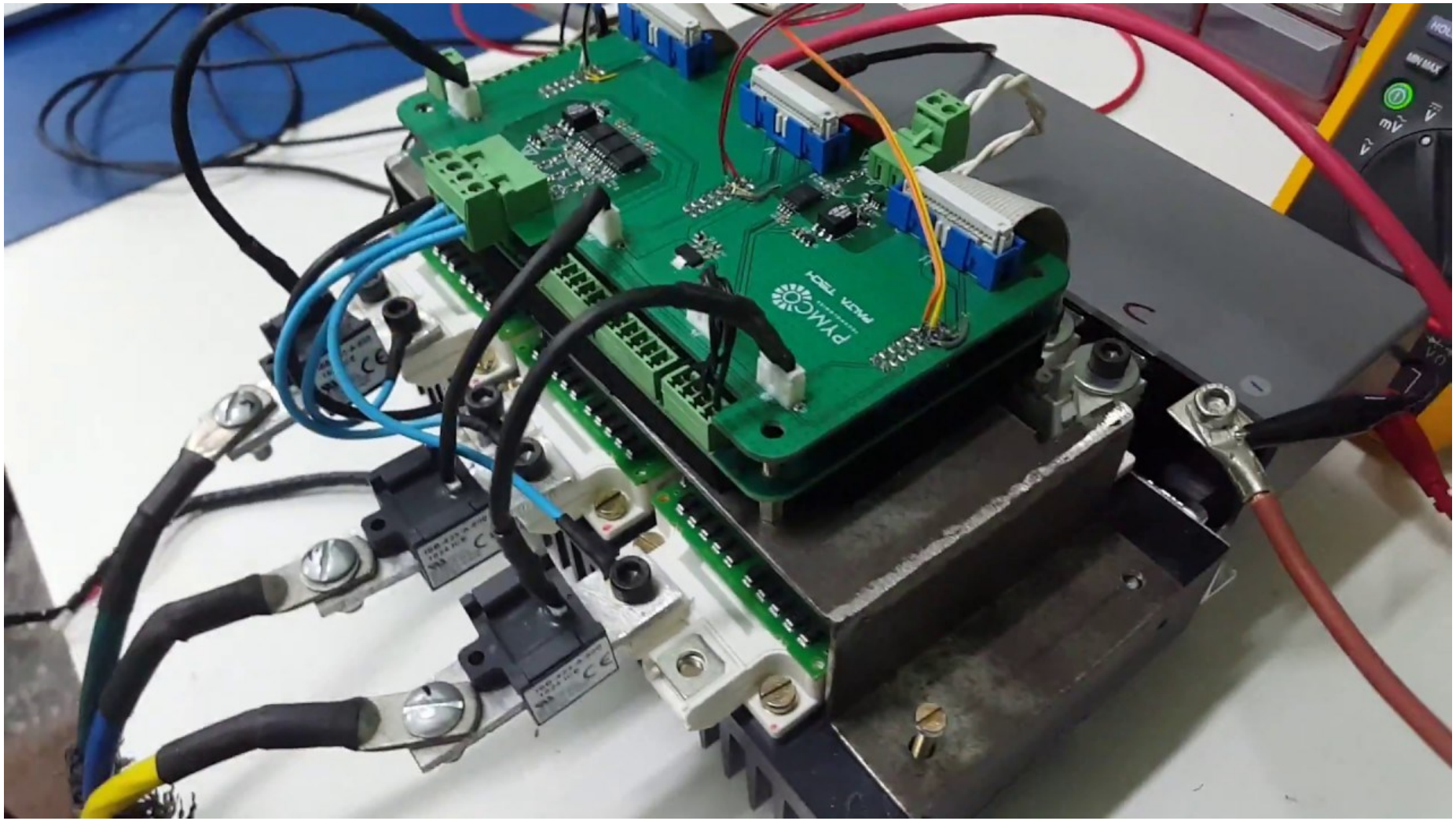
Mikrokontrolery i sterowniki cyfrowe



https://www.e-tronix.eu/2,sterownik-plc-programowalny-su-1-2.html

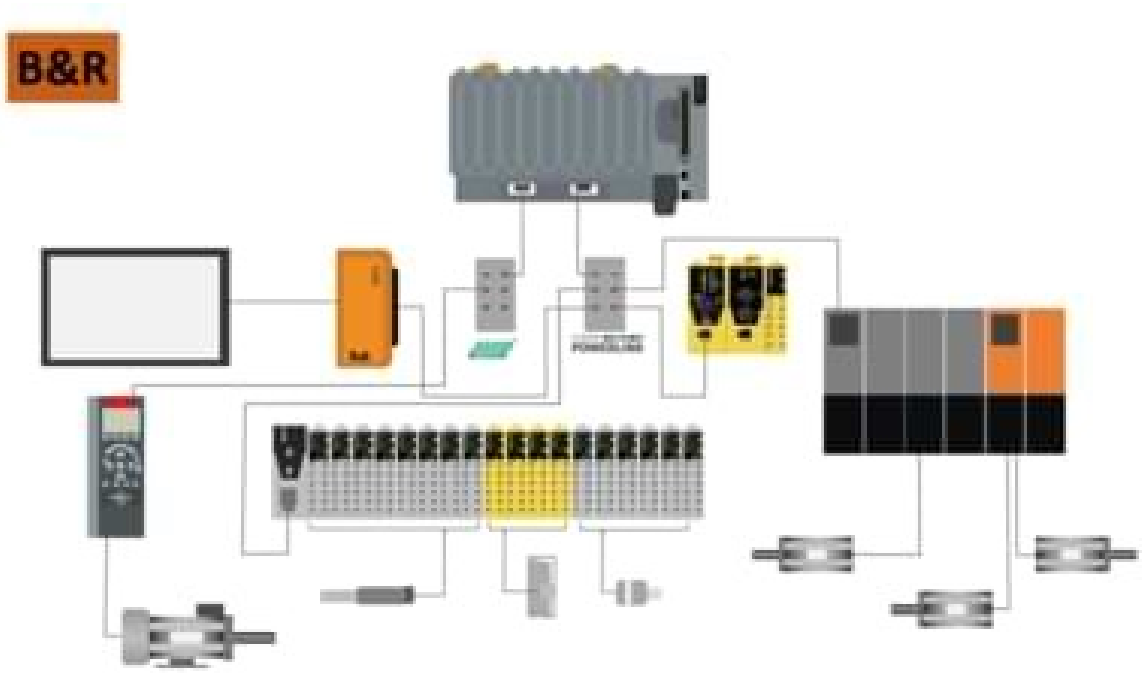
https://www.apar.pl/regulatory-uniwersalne.html

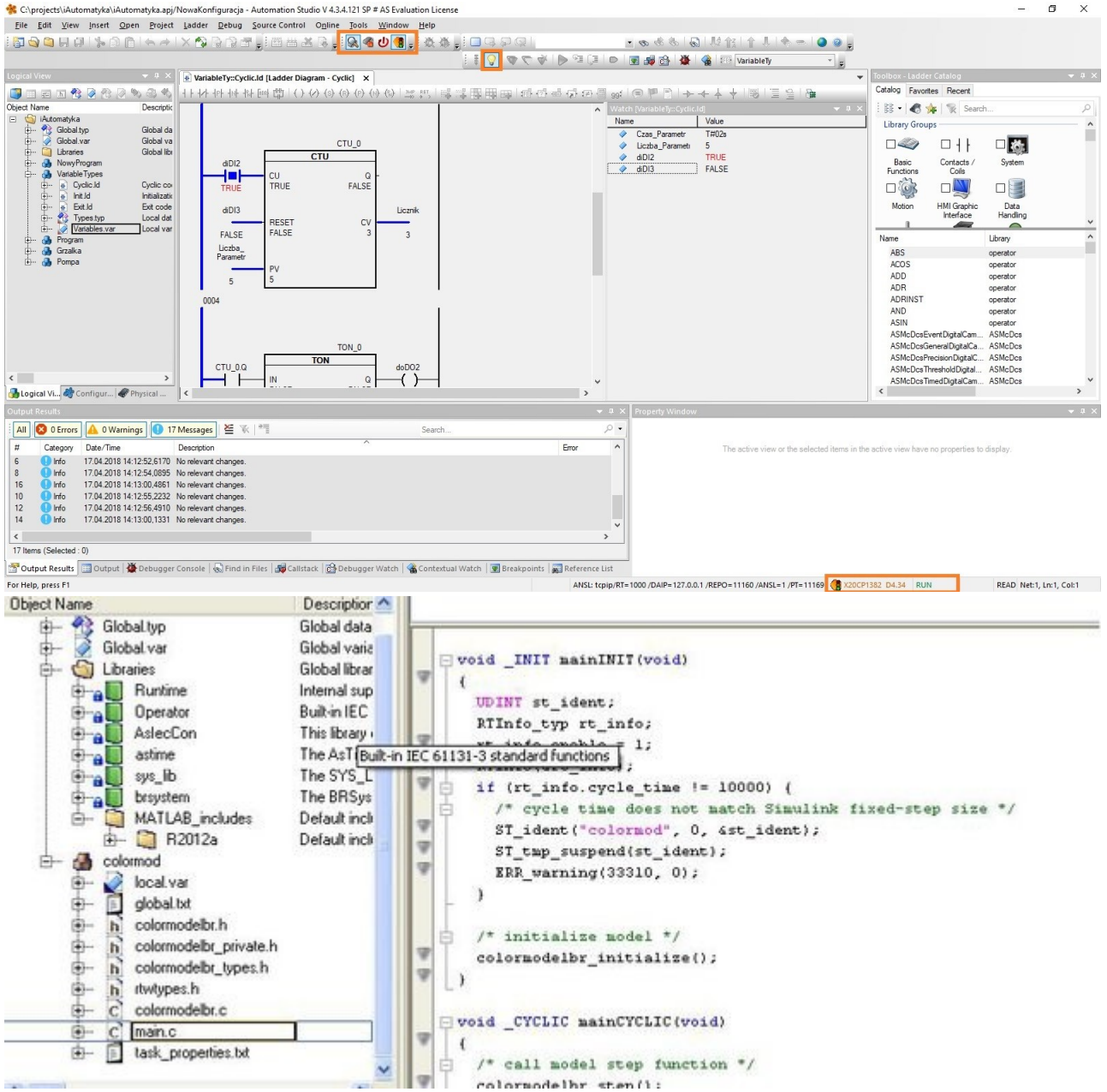
Sterowniki w energoelektronice



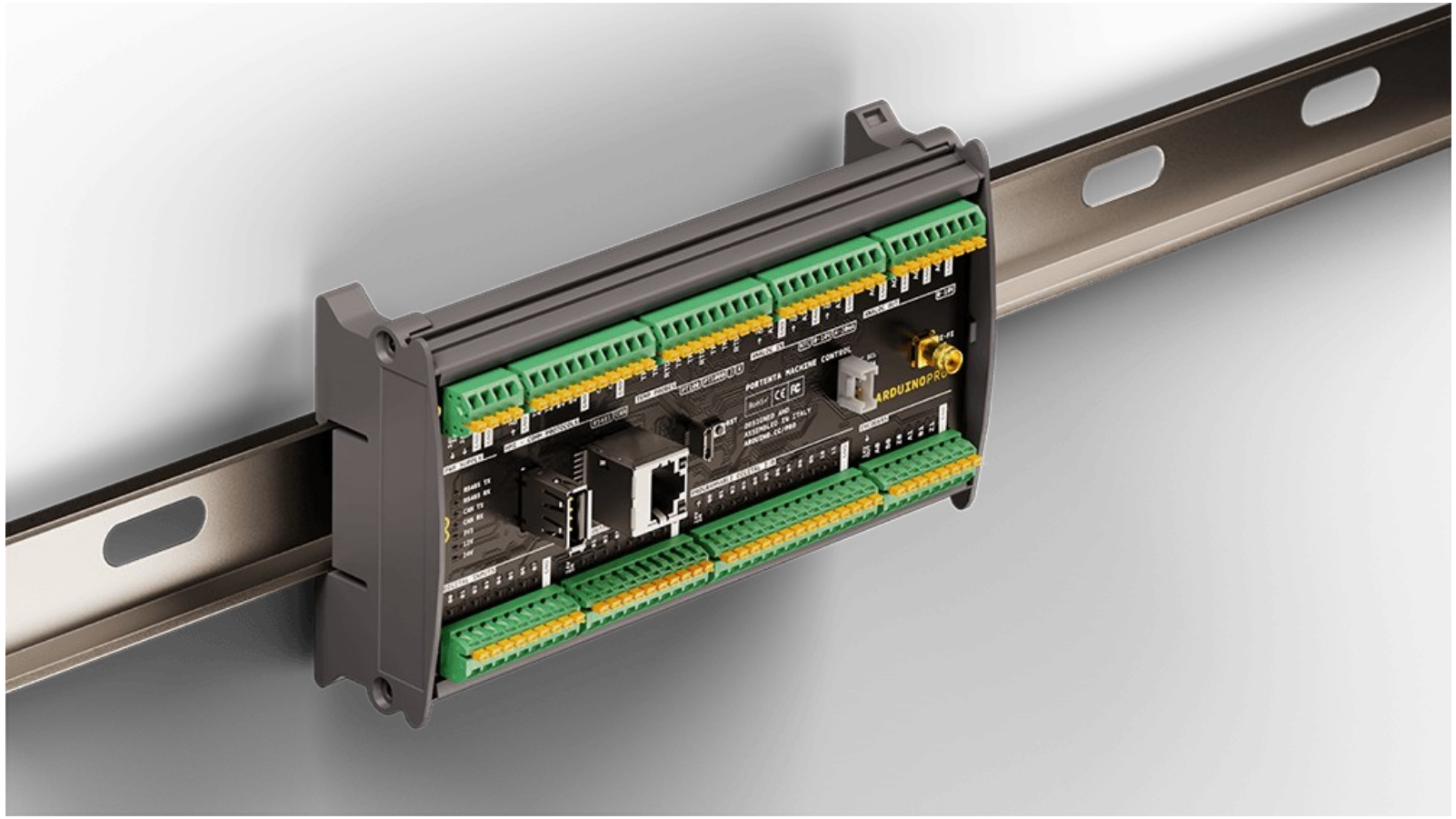
Programowanie sterowników

- Język drabinkowy
- Schematy blokowe
- Języki specjalizowane FBD, ST, SFC, LD
- **Język C/C++**

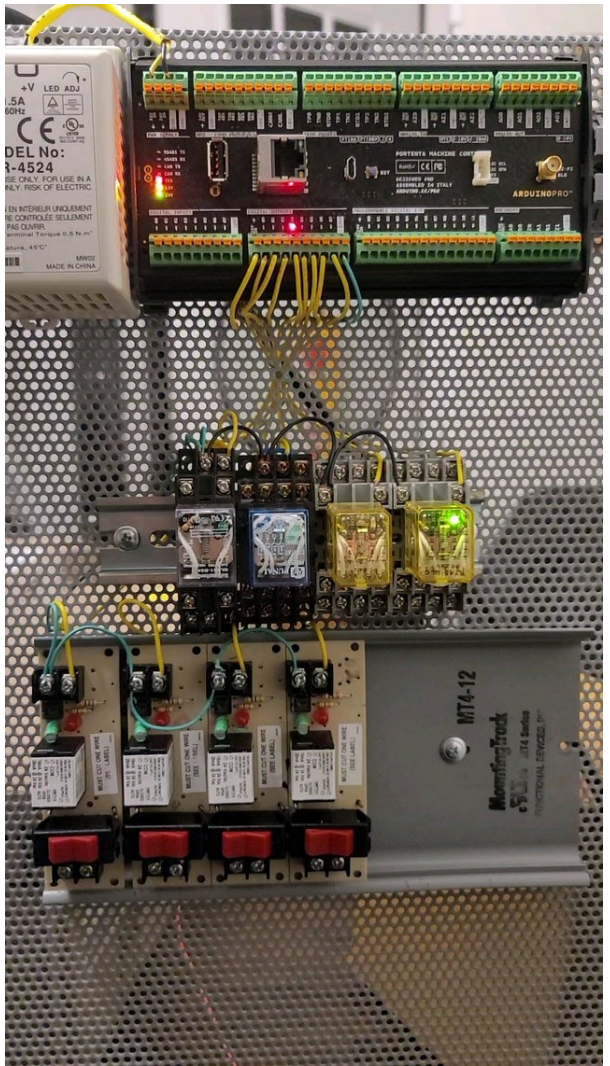




Arduino Portenta Machine Controller



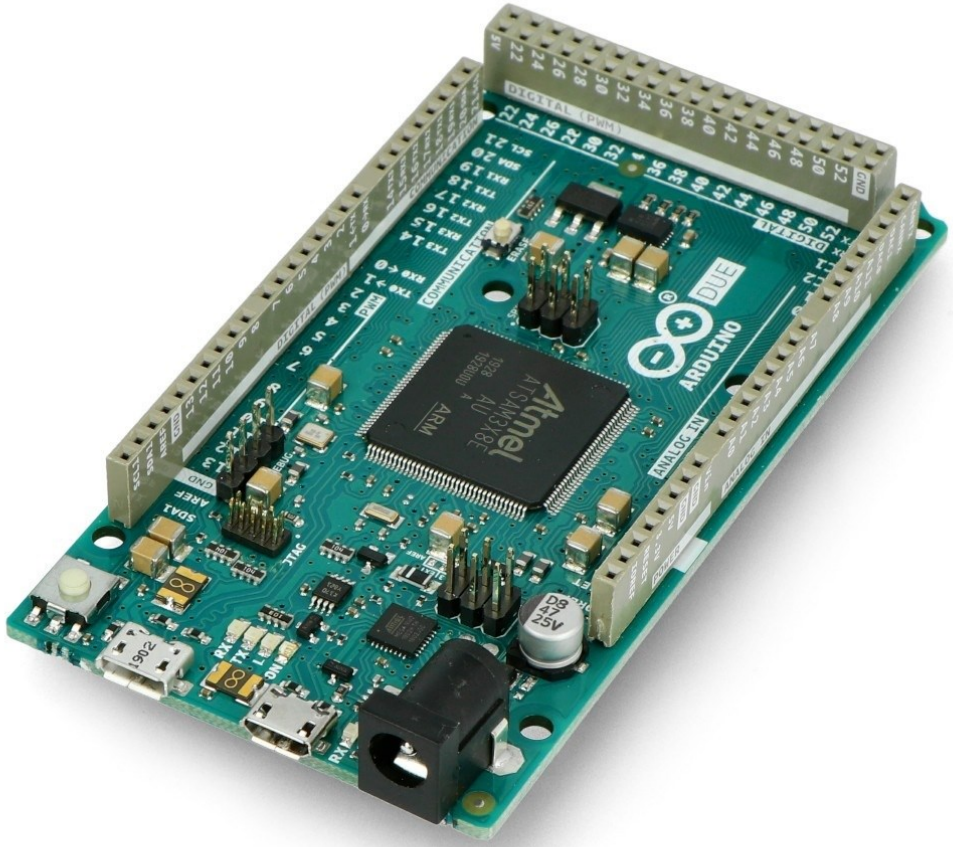
https://www.arduino.cc/pro/hardware/product/portenta-machine-control



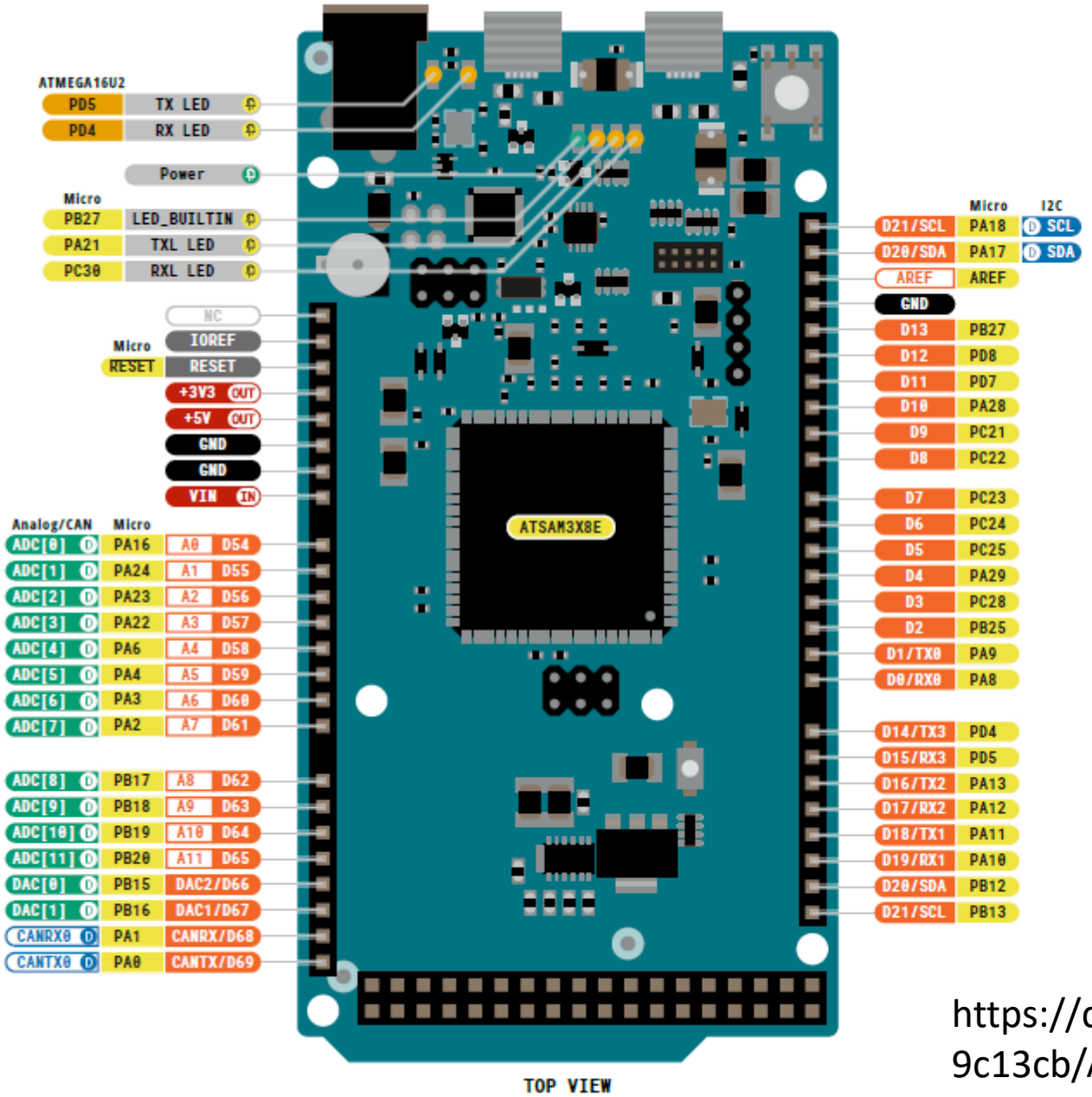
Plan ćwiczenia:

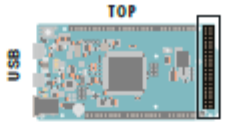
- Badanie procesu dyskretyzacji sygnału ciągłego
- Implementacja i walidacja dyskretnego odpowiednika układu ciągłego pierwszego rzędu
- Implementacja i walidacja dyskretnego odpowiednika układu ciągłego drugiego rzędu

Użyty w ćwiczeniu: Arduino Due

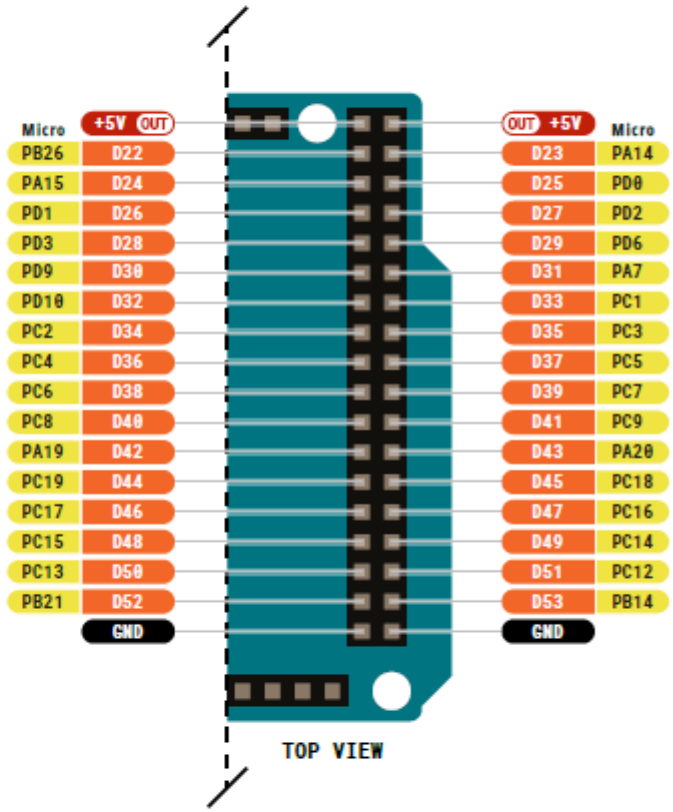


- Napięcie zasilania: od 7 V do 12 V
- Mikrokontroler: AT91 SAM3X8E rdzeń 32-bit
  - Maksymalna częstotliwość zegara: 84 MHz
  - Pamięć SRAM: 96 kB
  - Pamięć Flash: 512 kB
  - Piny I/O: 54
- Kanały PWM: 12
- Ilość wejść analogowych: 12 (kanały przetwornika A/C)
- Przetwornik C/A (cyfrowo-analogowy)
- Kontroler DMA
- Interfejsy szeregowe: UART, SPI, I2C, CAN, USB
- Debugger JTAG





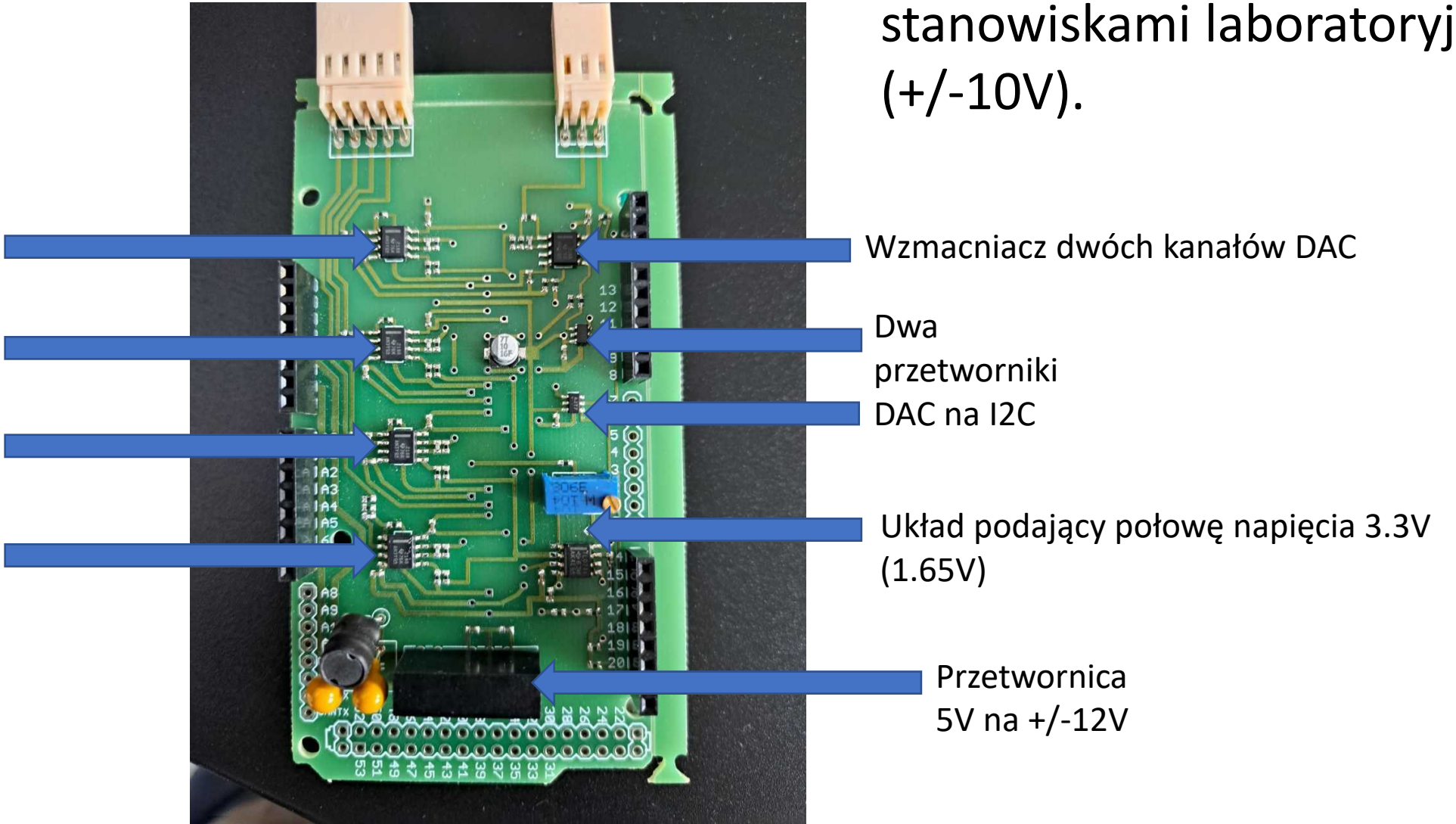
Digital pins D22-D53



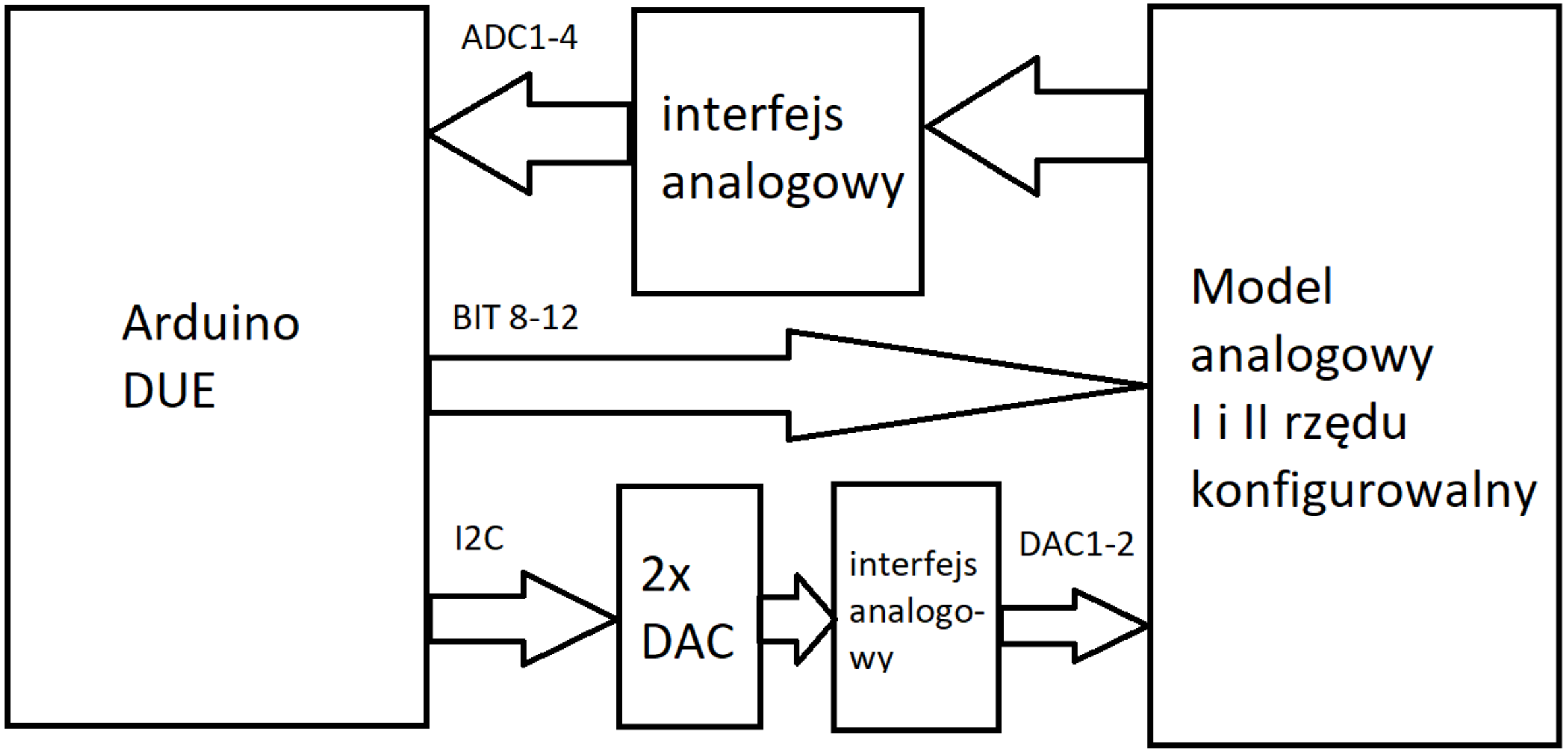
https://docs.arduino.cc/static/de243c00e7ed63603bae1c99519c13cb/A000056-full-pinout.pdf

Interfejs sterownika

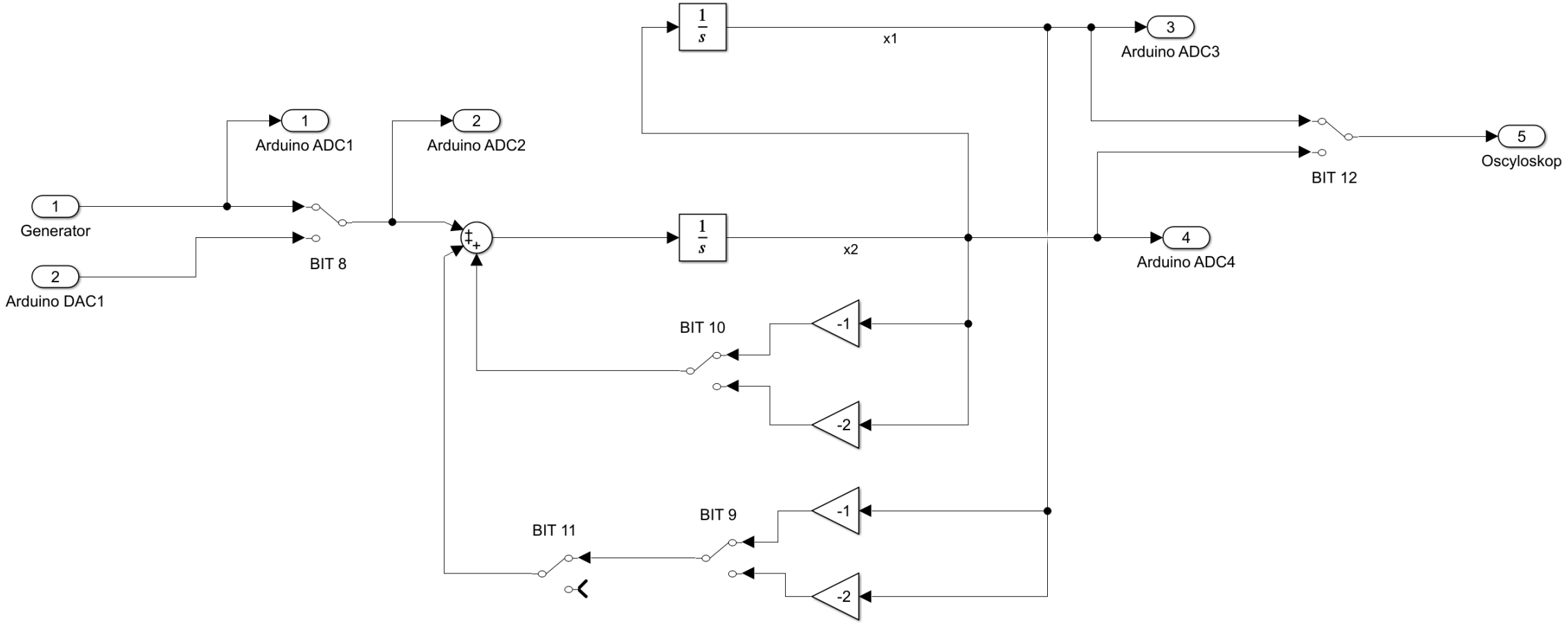
Zadanie: dopasowanie poziomów napięć pomiędzy płytką Arduino Due (0-3.3V) a stanowiskami laboratoryjnymi (+/-10V).



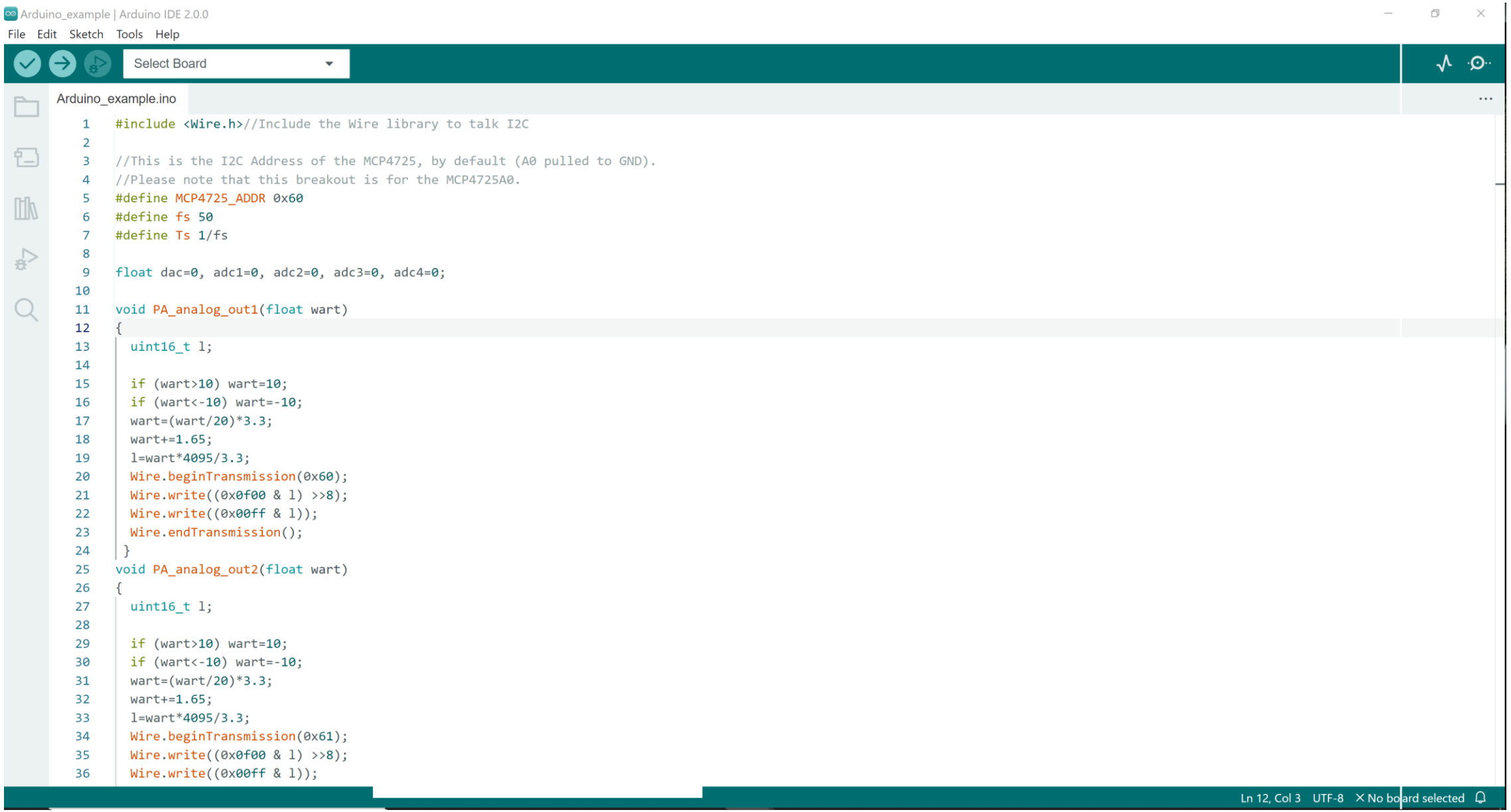
Schemat blokowy stanowiska laboratoryjnego



Schemat modelu analogowego I i II rzędu



Programowanie - Arduino IDE



```
42
43  void TC3_Handler()
44  {
45    adc1 = PA_analog_in(A0);
46    adc2 = PA_analog_in(A1);
47    adc3 = PA_analog_in(A2);
48    adc4 = PA_analog_in(A3);
49
50    dac = adc1;
51
52    PA_analog_out1(dac);
53
54    Serial.print(adc1);
55    Serial.print("\t");
56    Serial.print(adc2);
57    Serial.print("\t");
58    Serial.print(adc3);
59    Serial.print("\t");
60    Serial.println(adc4);
61
62    TC_GetStatus(TC1, 0);
63  }
64
65  void startTimer(Tc *tc, uint32_t channel, IRQn_Type irq, uint32_t frequency) {
66          pmc_set_writeprotect(false);
67          pmc_enable_periph_clk((uint32_t)irq);
68          TC_Configure(tc, channel, TC_CMR_WAVE | TC_CMR_WAVSEL_UP_RC | TC_CMR_TCCLKS_TIMER_CLOCK4);
69          uint32_t rc = VARIANT_MCK/128/frequency; //128 because we selected TIMER_CLOCK4 above
70          TC_SetRA(tc, channel, rc/2); //50% high, 50% low
71          TC_SetRC(tc, channel, rc);
72          TC_Start(tc, channel);
73          tc->TC_CHANNEL[channel].TC_IER=TC_IER_CPCS;
74          tc->TC_CHANNEL[channel].TC_IDR=~TC_IER_CPCS;
75          NVIC_EnableIRQ(irq);
76  }
77
```

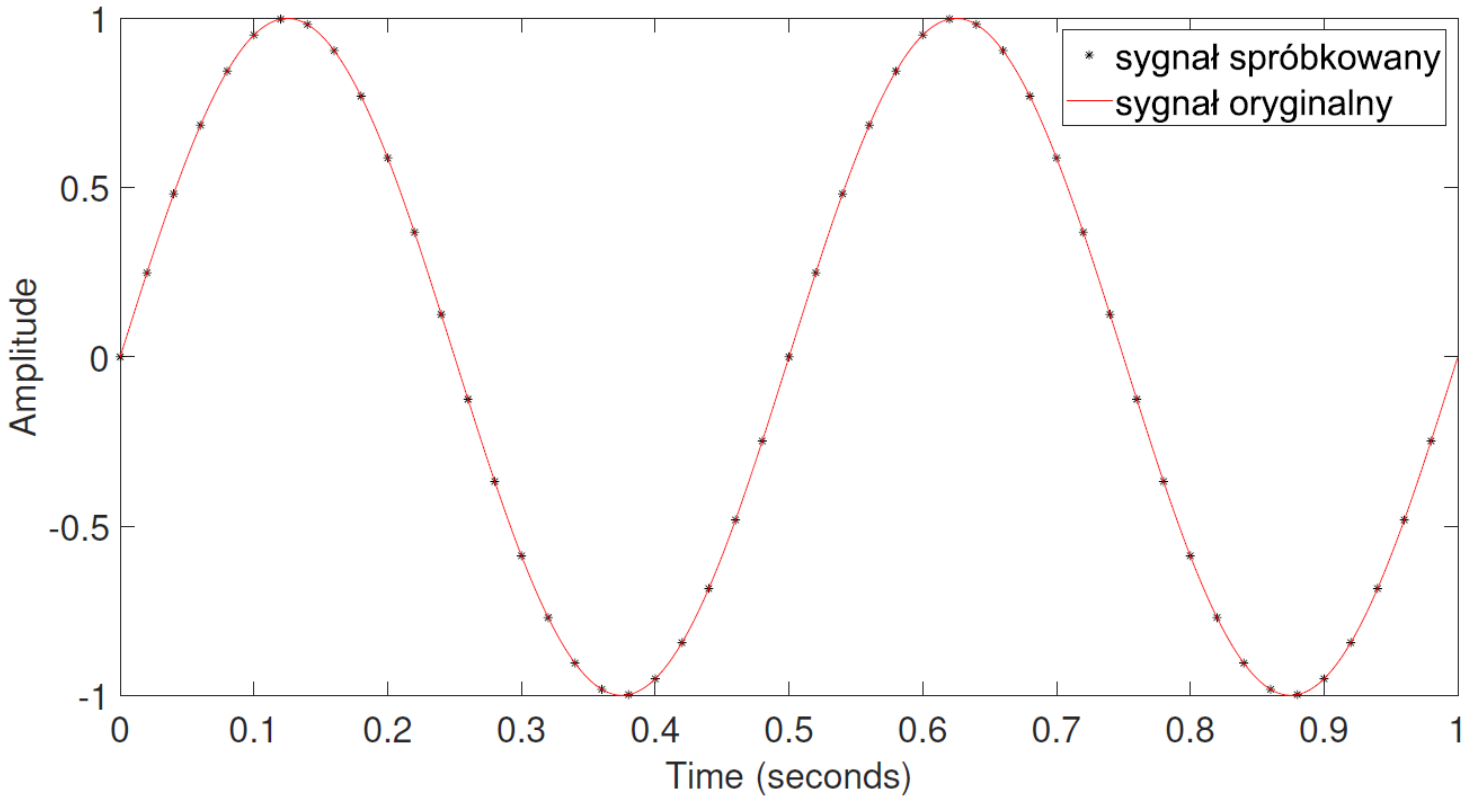
```
78  void setup()
79  {
80      analogReadResolution(12);
81      Wire.begin();
82      //Serial.begin(9600);
83      Serial.begin(115200);
84      pinMode(13,OUTPUT);
85
86      pinMode(8, OUTPUT);
87  pinMode(9, OUTPUT);
88  pinMode(10, OUTPUT);
89  pinMode(11, OUTPUT);
90  pinMode(12, OUTPUT);
91
92   digitalWrite(8, HIGH); /* 0 - sterowanie z generatora, 1- sterowanie z Arduino */
93  digitalWrite(9, LOW); /*A_1,2 0 - parametr -1, 1 - parametr -2 */
94  digitalWrite(10, LOW);/*A_2,2 0 - parametr -1, 1 - parametr -2 */
95  digitalWrite(11, LOW); /* 0 - rząd 2, 1 - rząd 1*/
96  digitalWrite(12, LOW); /* wyjście oscyloskopowe 0 - x_1 (rząd 2),  1 - x_2 (rząd 1)*/
97
98
99      startTimer(TC1, 0, TC3_IRQn, fs);
100 }
101 //------------------------------------------------
102 void loop()
103 {
104
105 }
106
```

Badanie procesu dyskretyzacji sygnału ciągłego

- Twierdzenie o próbkowaniu (Shannona vel Shanonna-Nyquista):

Jeśli sygnał ciągły nie ma składowych widma o częstotliwości równej lub większej niż B, może on zostać wiernie odtworzony z ciągu jego próbek tworzących sygnał dyskretny, o ile próbki te zostały pobrane z częstotliwością co najmniej 2B.

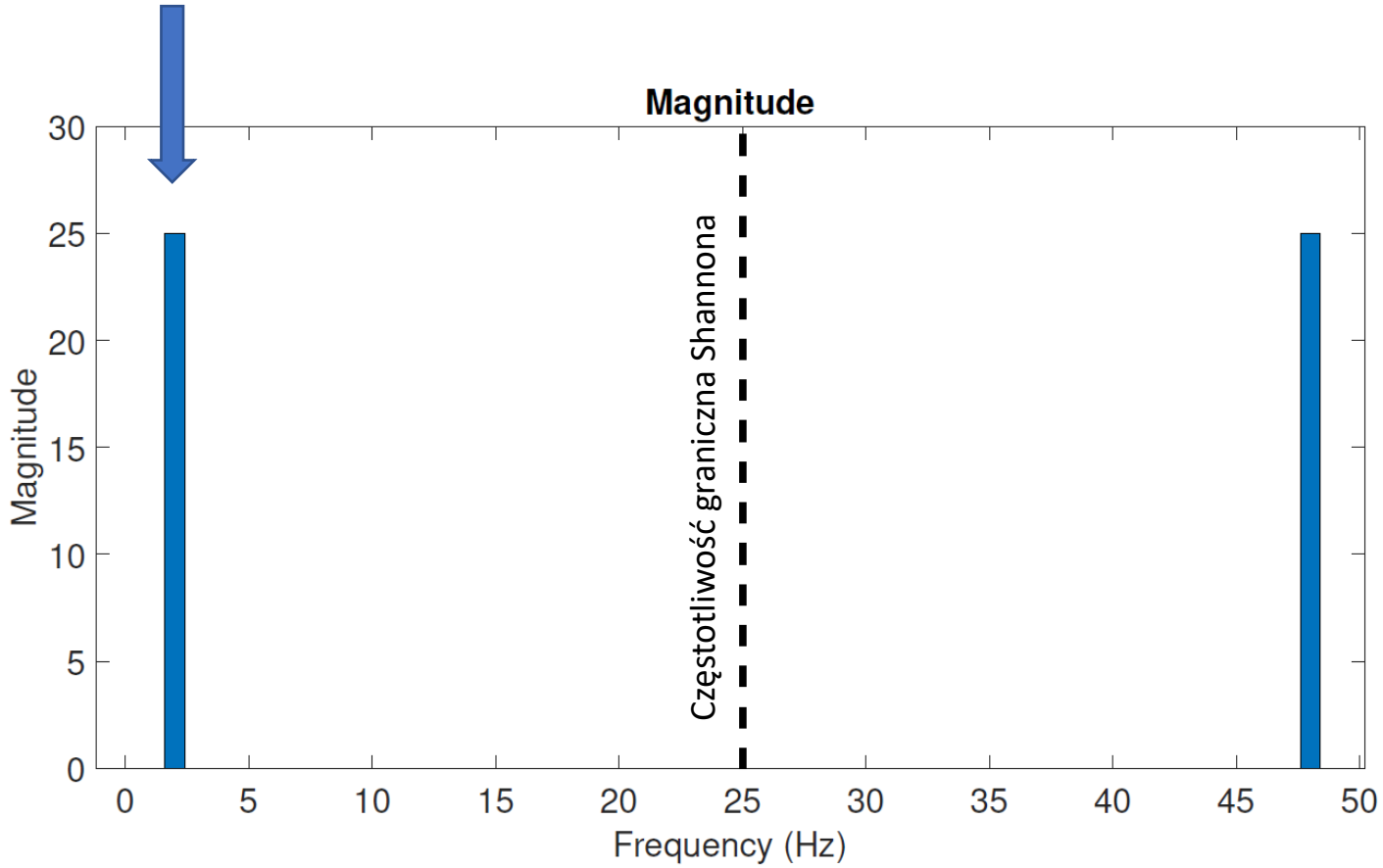
Próbkowanie sygnału 2Hz



Rysunek 1. Próbkowanie sygnału 2 Hz z częstotliwością próbkowania 50Hz

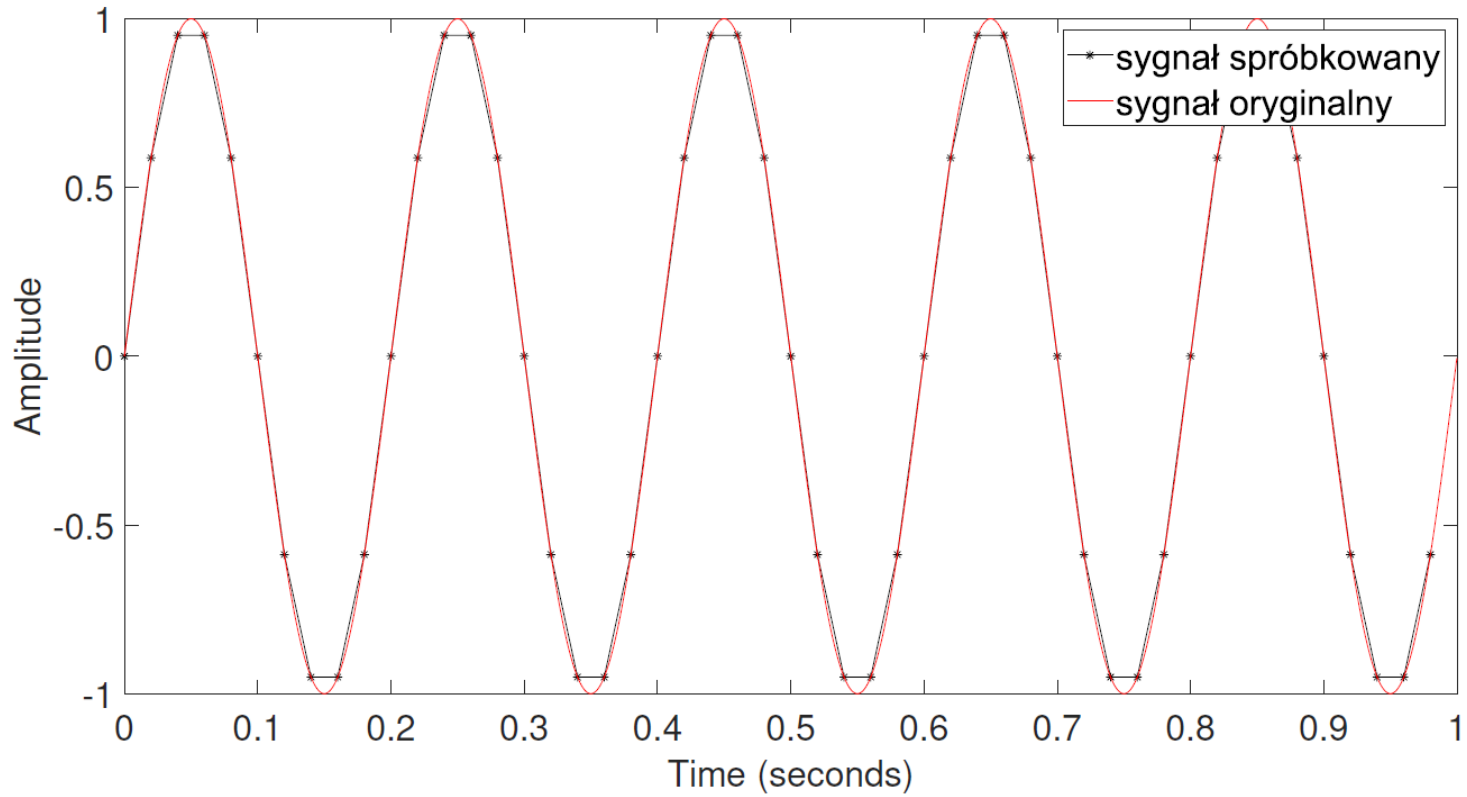
Próbkowanie sygnału 2Hz

amplituda_sygnału=2*Magnitude/L



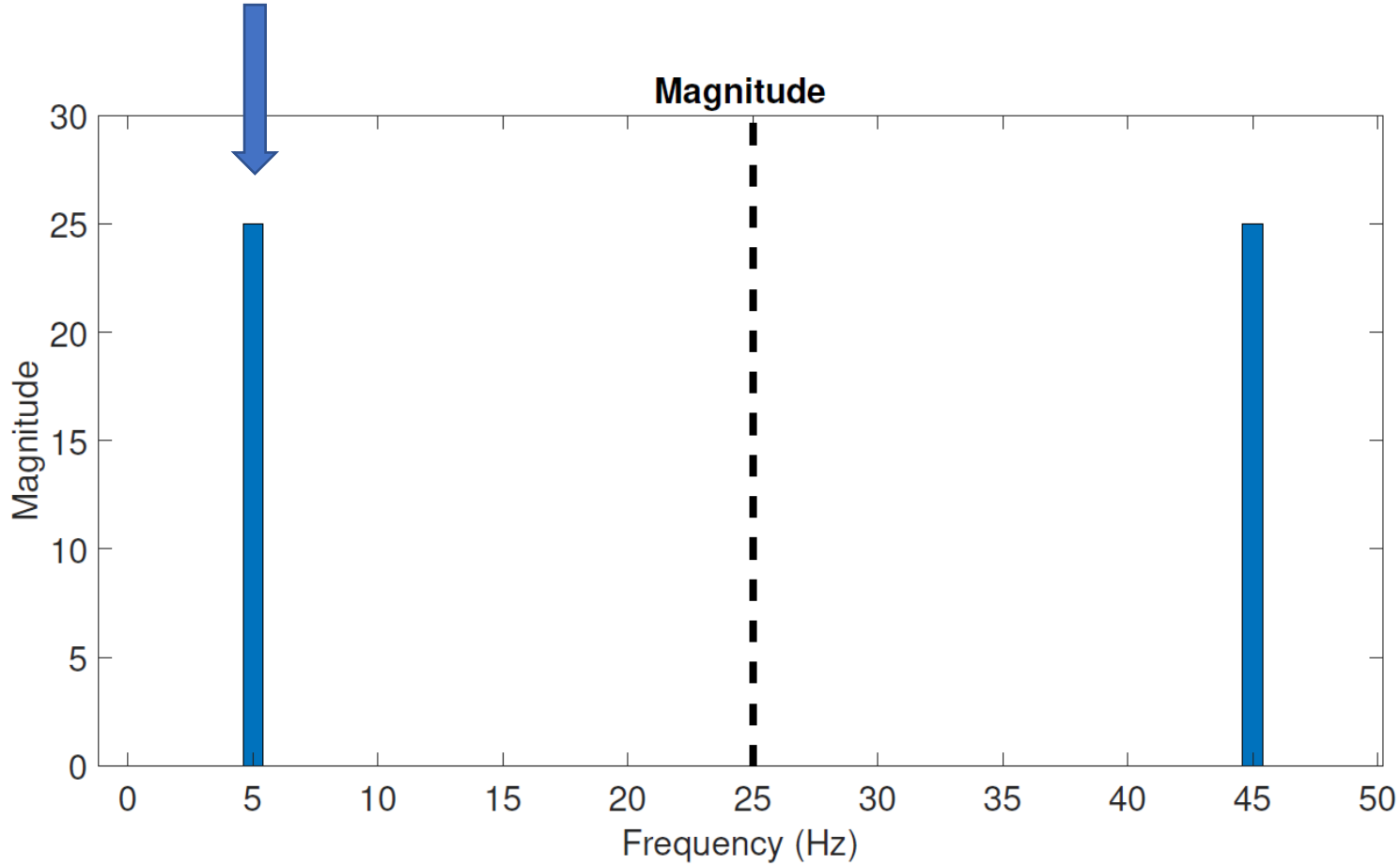
Rysunek 2. FFT próbkowania sygnału 2 Hz z częstotliwością próbkowania 50Hz

Próbkowanie sygnału 5Hz



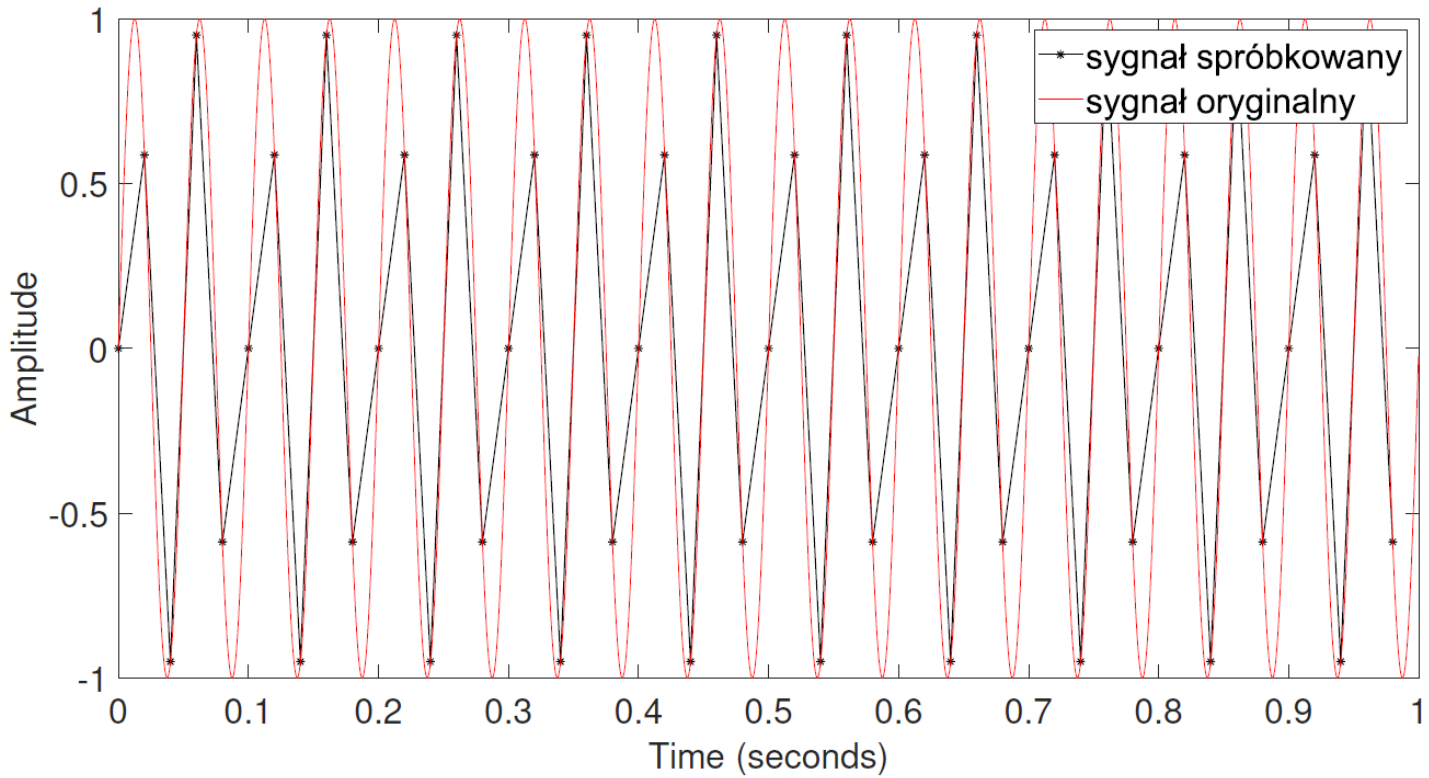
Rysunek 3. Próbkowanie sygnału 5 Hz z częstotliwością próbkowania 50Hz

Próbkowanie sygnału 5Hz



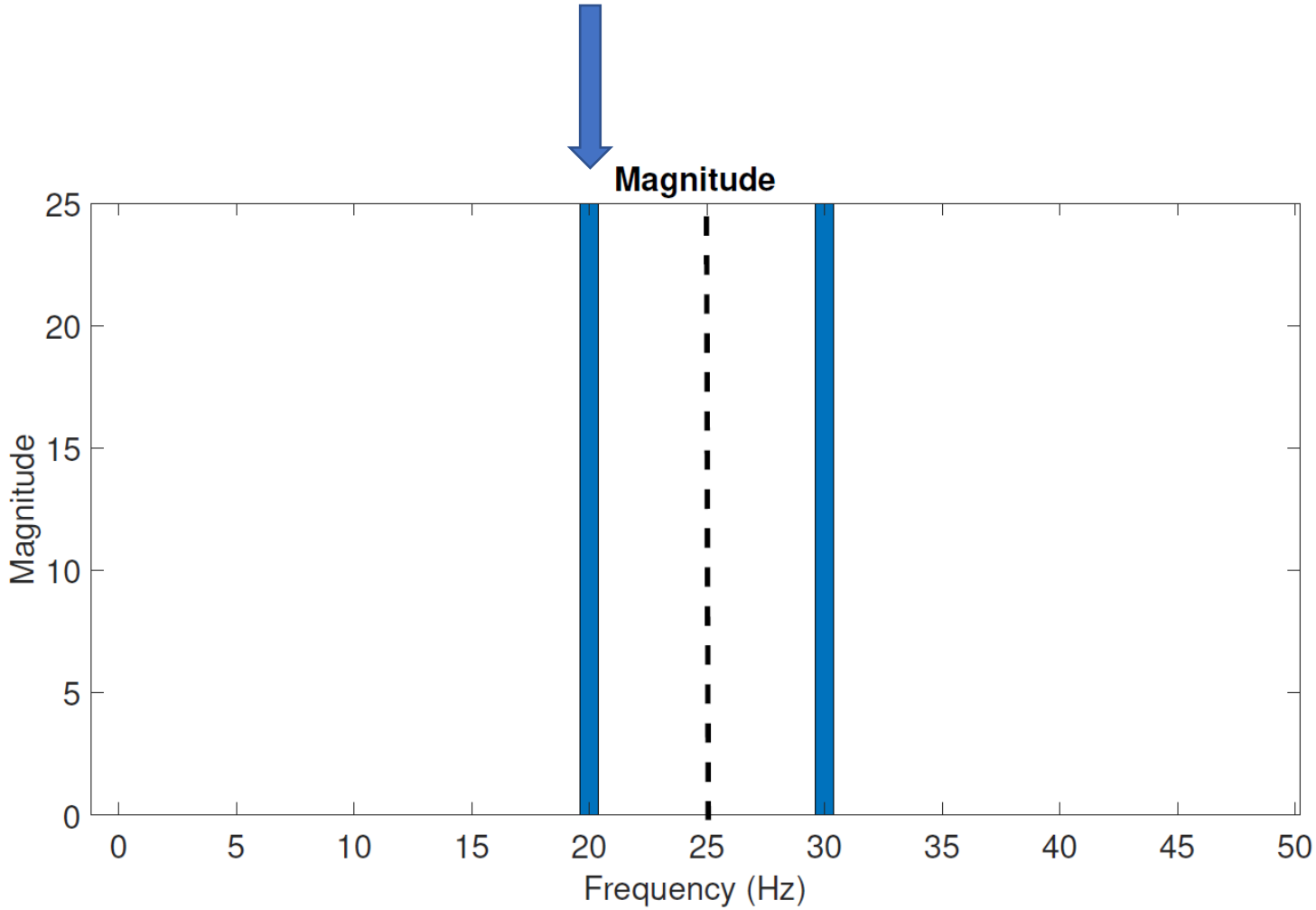
Rysunek 4. FFT próbkowania sygnału 5 Hz z częstotliwością próbkowania 50Hz

Próbkowanie sygnału 20Hz



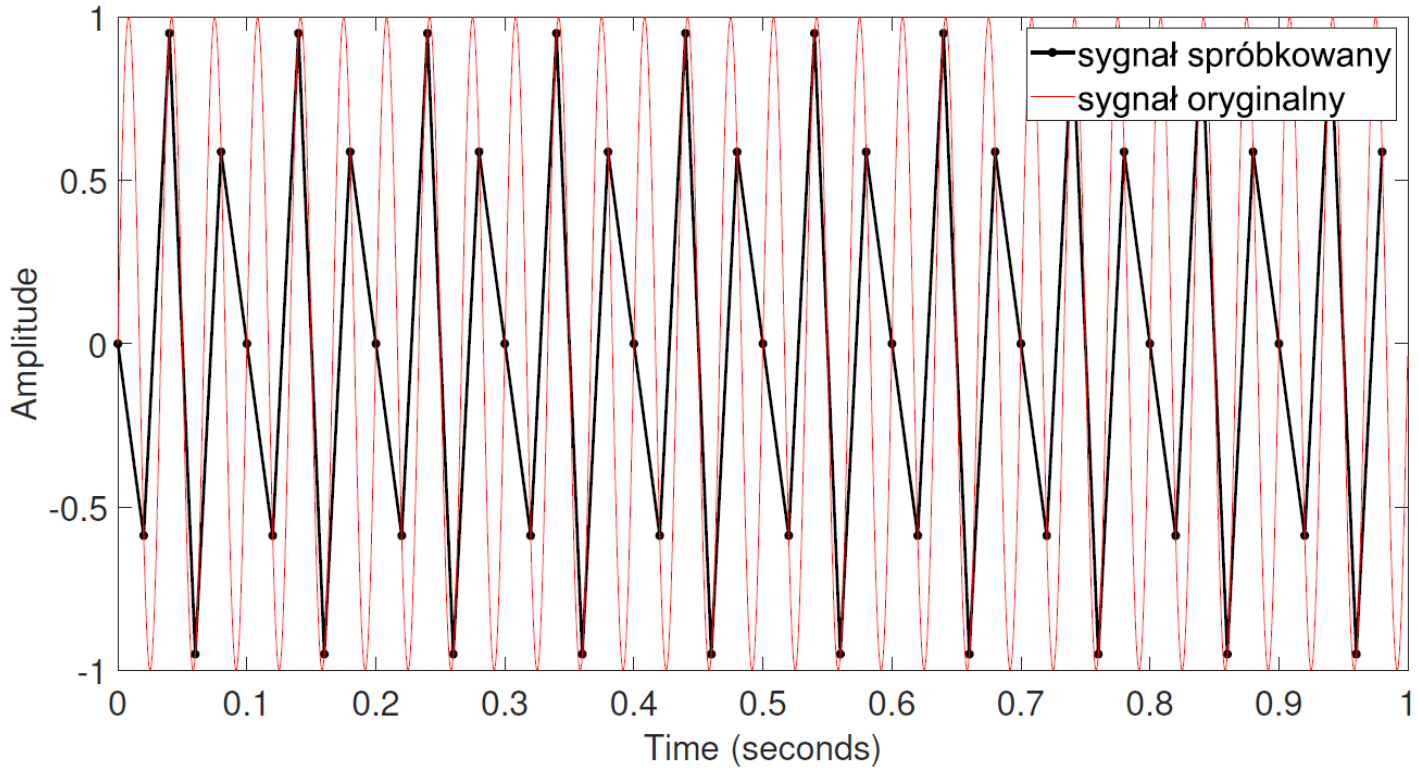
Rysunek 5. Próbkowanie sygnału 20 Hz z częstotliwością próbkowania 50Hz

Próbkowanie sygnału 20Hz



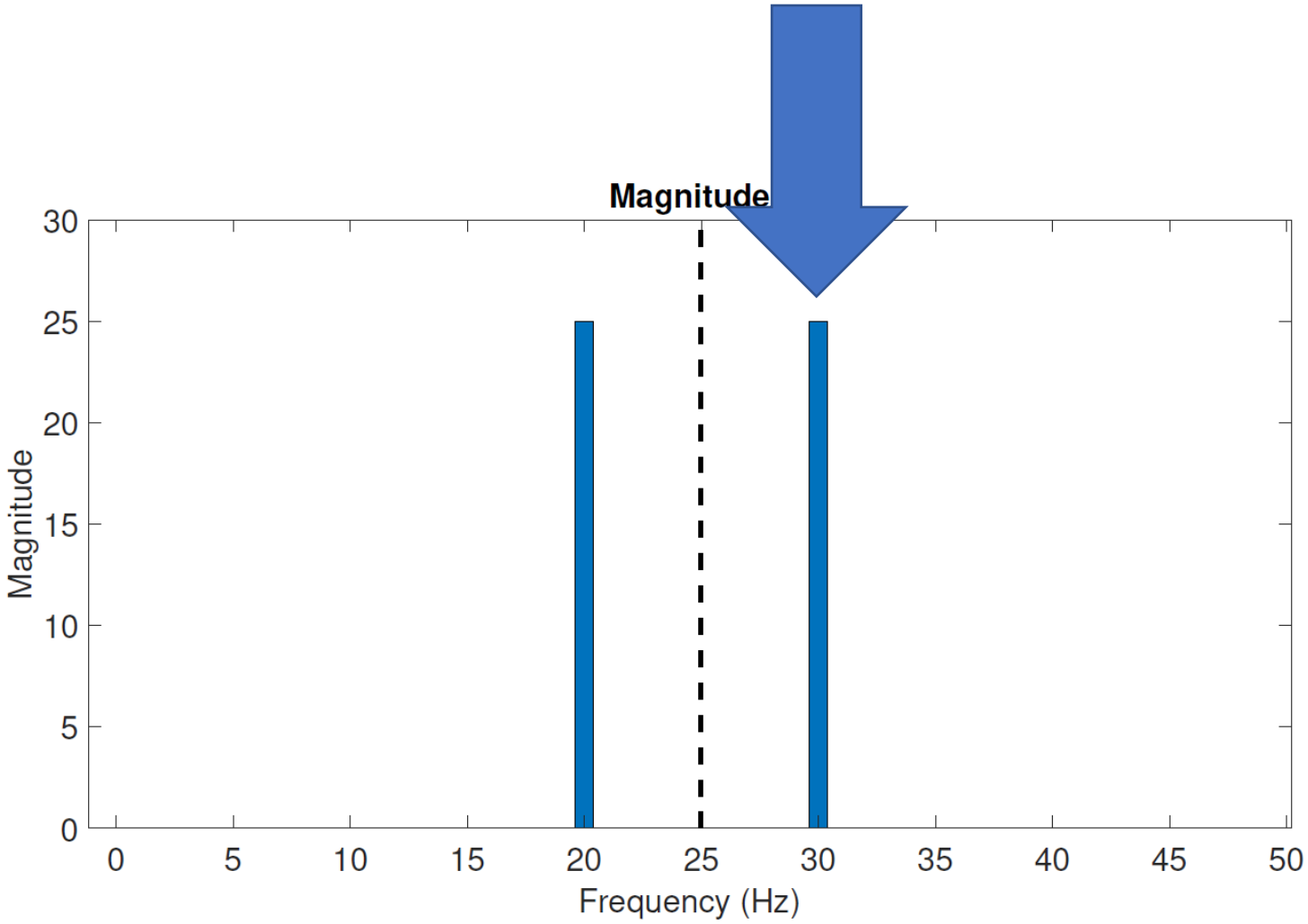
Rysunek 6. FFT próbkowania sygnału 20 Hz z częstotliwością próbkowania 50Hz

Próbkowanie sygnału 30Hz



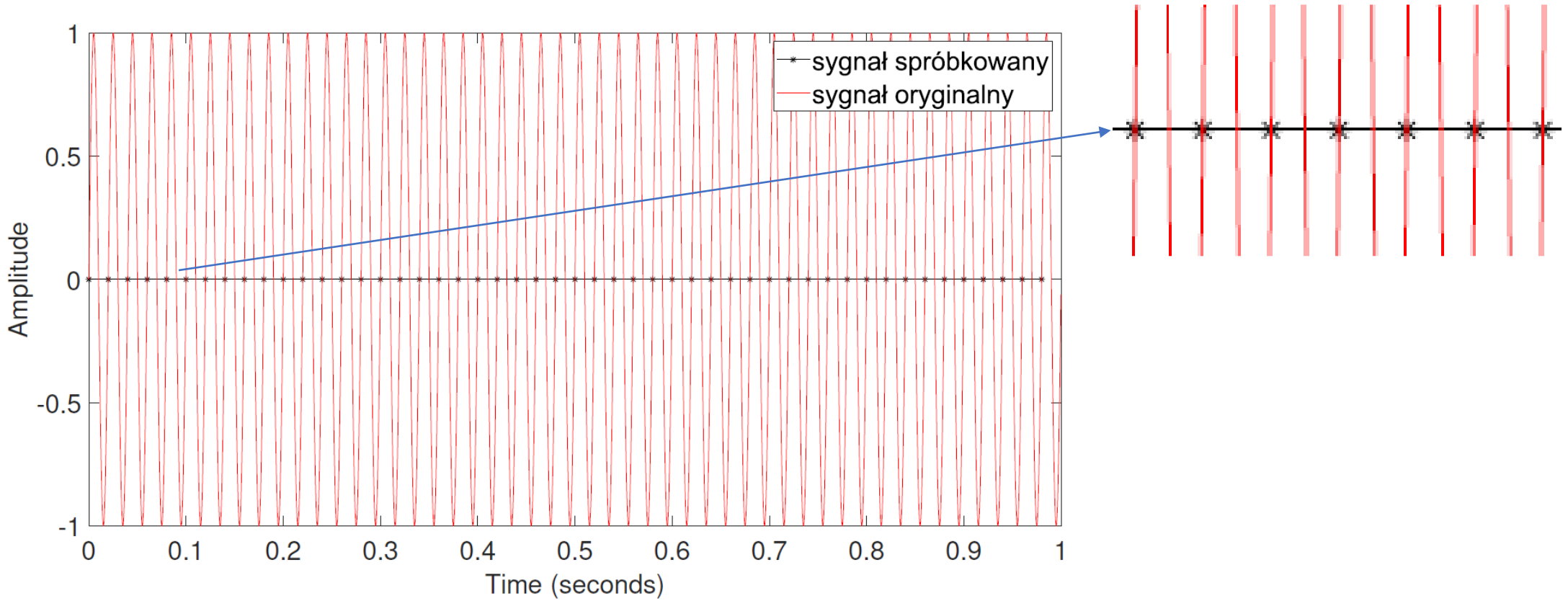
Rysunek 7. Próbkowanie sygnału 30 Hz z częstotliwością próbkowania 50Hz

Próbkowanie sygnału 30Hz



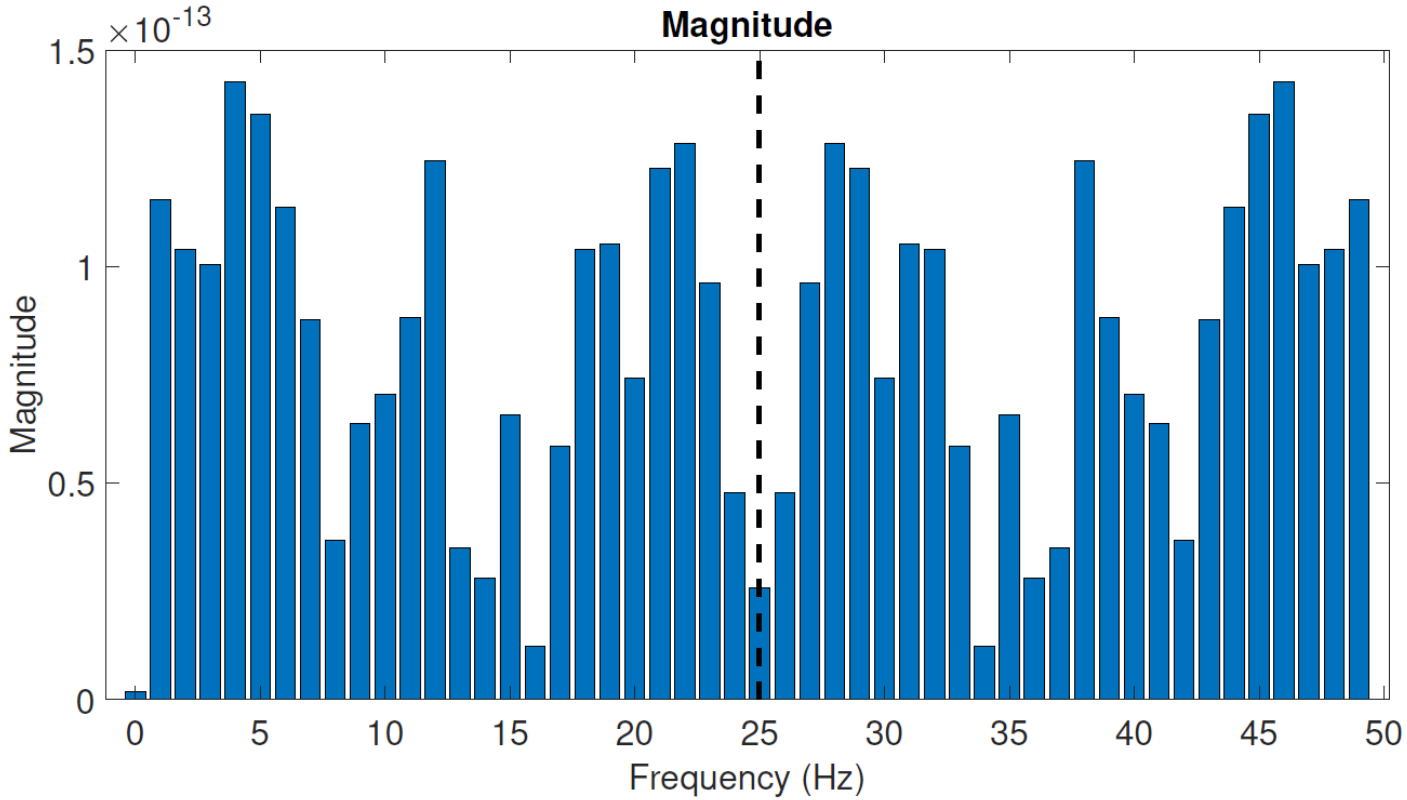
Rysunek 8. FFT próbkowania sygnału 30 Hz z częstotliwością próbkowania 50Hz

Próbkowanie sygnału 50Hz



Rysunek 9. Próbkowanie sygnału 50 Hz z częstotliwością próbkowania 50Hz

Próbkowanie sygnału 50Hz



Rysunek 10. FFT próbkowania sygnału 50 Hz z częstotliwością próbkowania 50Hz

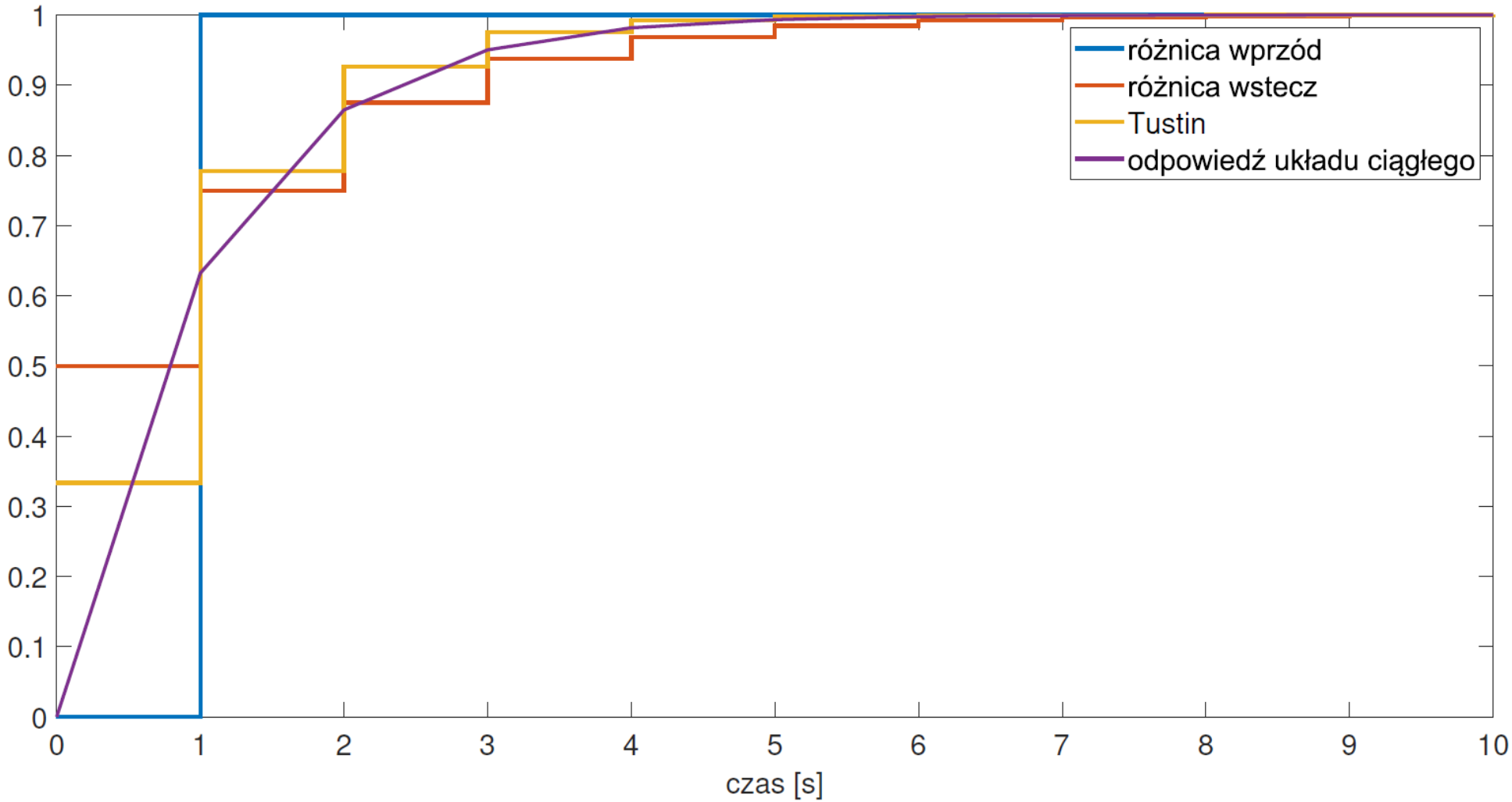
Badania w laboratorium

Obserwacje przebiegów dla różnych częstotliwości sygnałów wymuszających, ze szczególnym uwzględnieniem granicy Shannon’a oraz częstotliwości sygnału równej bądź bliskiej częstotliwości próbkowania

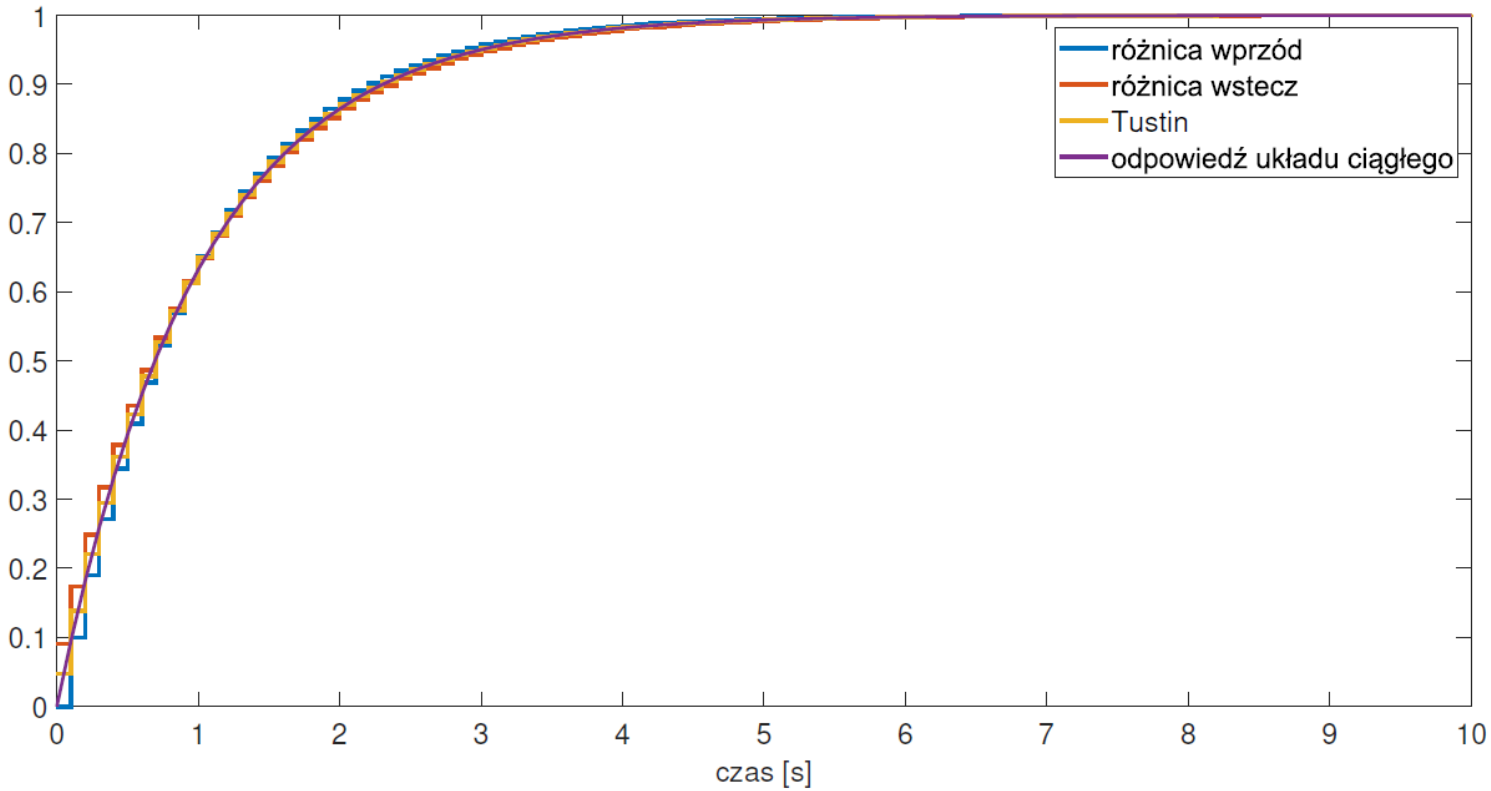
Implementacja układów dyskretnych

Rodzaje dyskretyzacji:

- Aproksymacji pochodnej:
  - Różnicy wprzód $\frac{\mathrm{d}y(t)}{\mathrm{d}t} \approx \Delta y_k = \frac{y_{k+1} - y_k}{T}$
  - Różnicy w tył $\frac{\mathrm{d}y(t)}{\mathrm{d}t} \approx \Delta y = \frac{y_k - y_{k-1}}{T}$
  - Tustina $c_{k+1} = c_k + \frac{T}{2}(u_k + u_{k+1})$



Rysunek. Porównanie odpowiedzi czasowych układu ciągłego i jego dyskretyzacji metodami aproksymacji pochodnej dla okresu próbkowania $T = 1$



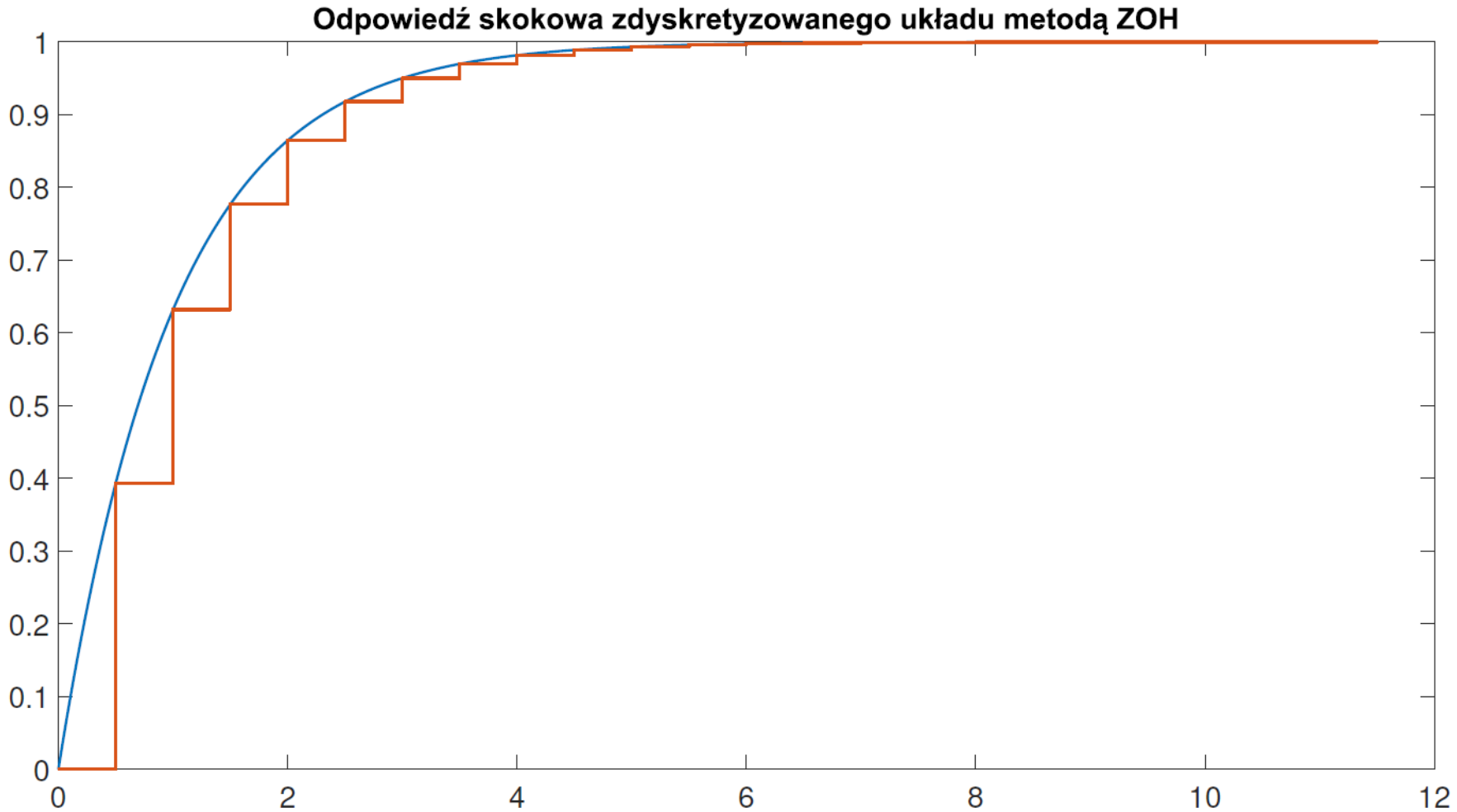
Rysunek. Porównanie odpowiedzi czasowych układu ciągłego i jego dyskretyzacji metodami aproksymacji pochodnej dla okresu próbkowania $T = 0.1$

Implementacja układów dyskretnych (2)

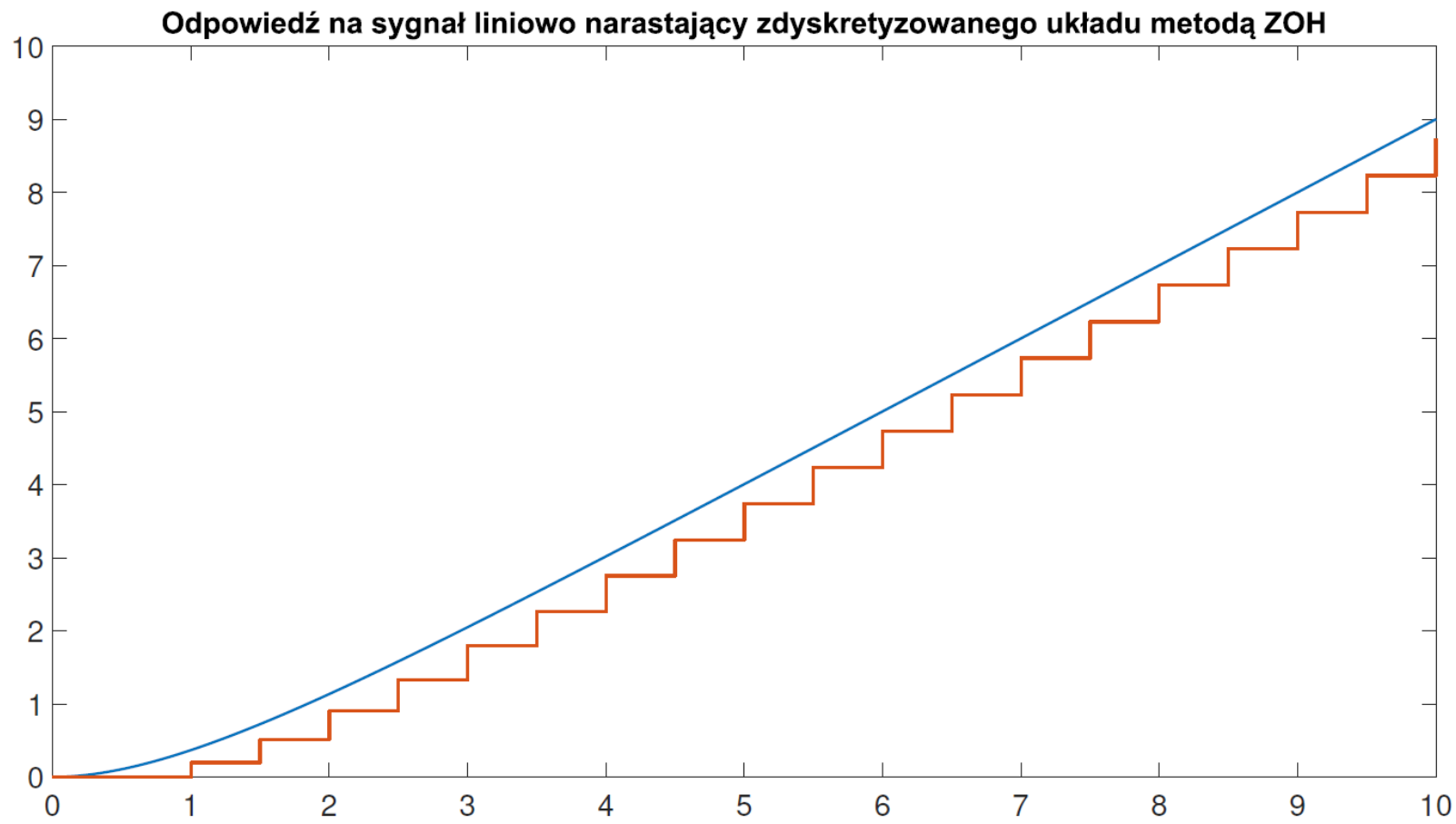
- Rodzaje dyskretyzacji cd. :

Odpowiednika odpowiedzi:

- Impulsowej $g(t) = \mathcal{L}^{-1}\{G(s)\}$ $g_k = g(Tk)$ $G(z) = \mathcal{Z}\{g_k\}$
- skokowej (ZOH)
- Liniowo-narastającej (FOH)



Rysunek. Odpowiedź skokowa zdyskretyzowanego układu metodą ZOH



Rysunek. Odpowiedź na sygnał liniowo narastający zdyskretyzowanego układu metodą ZOH

Dyskretyzacja w Matlabie

SYSD = c2d(SYSC,TS,METHOD) computes a discrete-time model SYSD with
sample time TS that approximates the continuous-time model SYSC.
The string METHOD selects the discretization method among the following:
'zoh' Zero-order hold on the inputs
'foh' Linear interpolation of inputs
'impulse' Impulse-invariant discretization
'tustin' Bilinear (Tustin) approximation.
'matched' Matched pole-zero method (for SISO systems only).
The default is 'zoh' when METHOD is omitted. The sample time TS should be specified in the time units of SYSC (see "TimeUnit" property).

Transmitancja dyskretna

Definicja transmitancji

$$G(z) = \frac{Y(z)}{U(z)}.$$

oczywiście przy zerowych warunkach początkowych.
Transmitancję dyskretną można przedstawić w postaci ogólnej będącej ilorazem wielomianów względem zmiennej z

$$G(z) = \frac{b_m z^m + b_{m-1} z^{m-1} + \cdots + b_1 z + b_0}{a_n z^n + a_{n-1} z^{n-1} + \cdots + a_1 z + a_0}. \tag{10}$$

z transmitancji

$$\frac{Y(z)}{U(z)} = \frac{b_m z^m + b_{m-1} z^{m-1} + \cdots + b_1 z + b_0}{a_n z^n + a_{n-1} z^{n-1} + \cdots + a_1 z + a_0}.$$

$$U(z)(b_m z^m + b_{m-1} z^{m-1} + \cdots + b_1 z + b_0) = Y(z)(a_n z^n + a_{n-1} z^{n-1} + \cdots + a_1 z + a_0)$$

korzystając z zerowych warunków początkowych i własności transformaty $\mathcal{Z}$

$$\mathcal{Z}\{f_{k+1}\} = zF(z) - zf_0$$

$$\mathcal{Z}\{f_{k+n}\} = z^n F(z) - \sum_{i=0}^{n-1} z^{n-i} f_i = z^n F(z) - z^n f_0 - z^{n-1} f_1 - \cdots - z f_{n-1}$$

możemy otrzymać

$$\begin{aligned} & a_n y_{k+n} + a_{n-1} y_{k+n-1} + \cdots + a_1 y_{k+1} + a_0 y_k \\ = \quad & b_m u_{k+m} + b_{m-1} u_{k+m-1} + \cdots + b_1 u_{k+1} + b_0 u_k \end{aligned}$$

Czyli równanie różnicowe odpowiadające danej transmitancji

Równanie różnicowe

W dziedzinie dyskretnego czasu dynamikę opisujemy równaniami różnicowymi

$$\begin{aligned} & a_n y_{k+n} + a_{n-1} y_{k+n-1} + \cdots + a_1 y_{k+1} + a_0 y_k \\ = \quad & b_m u_{k+m} + b_{m-1} u_{k+m-1} + \cdots + b_1 u_{k+1} + b_0 u_k, \end{aligned}$$

które można przekształcić

$$\begin{aligned} y_{k+n} \quad = \quad & \frac{1}{a_n} \left(b_m u_{k+m} + b_{m-1} u_{k+m-1} + \cdots + b_1 u_{k+1} + b_0 u_k \right. \\ - \quad & \left. a_{n-1} y_{k+n-1} - \cdots - a_1 y_{k+1} - a_0 y_k \right). \end{aligned}$$

np: $y_k = u_k - 0.5y_{k-1} + 0.7y_{k-2}$ odpowiedź przy zerowych warunkach początkowych i skoku jednostkowego

$$y_1 = u_1 - 0.5y_0 = 1$$

$$y_2 = u_2 - 0.5y_1 + 0.7y_0 = 0.5$$

$$y_3 = u_3 - 0.5y_2 + 0.7y_1 = 1 - 0.5 + 0.7 = 1.2$$

Przykład implementacji układu 1-go rzędu

Weźmy układ ciągły pierwszego rzędu dany następującą transmitancją

$$G(s) = \frac{1}{s+1}$$

dokonajmy jej dyskretyzacji w Matlabie

```
>> sys=tf([1],[1 1])

sys =

   1
  -----
  s + 1

Continuous-time transfer function.
```

```
>> sysd=c2d(sys,1/50)

sysd =

   0.0198
  ----------
  z - 0.9802

Sample time: 0.02 seconds
Discrete-time transfer function.
```

Otrzymujemy więc transmitancję dyskretną w następującej postaci

$$\frac{Y(z)}{U(z)} = \frac{0.0198}{z - 0.9802} \tag{21}$$

co przekształcamy do postaci

$$Y(z)(z - 0.9802) = U(z)(0.0198). \tag{22}$$

korzystając z zerowych warunków początkowych i własności transformaty $\mathcal{Z}$ otrzymujemy następującą postać równania różnicowego

$$y_{k+1} - 0.9802y_k \quad = \quad 0.0198u_k$$

co możemy przekształcić

$$y_{k+1} \quad = \quad 0.0198u_k + 0.9802y_k$$

co jest równoważne

$$y_k \quad = \quad 0.0198u_{k-1} + 0.9802y_{k-1}$$

Implementacja na arduino

$$y_k = 0.0198u_{k-1} + 0.9802y_{k-1}$$

```
void TC3_Handler()
{
  adc1 = PA_analog_in(A0);
  adc2 = PA_analog_in(A1);
  adc3 = PA_analog_in(A2);
  adc4 = PA_analog_in(A3);

  yk=  0.0198*ukp+0.9802*yk;
  ukp=adc1;
```

```
  Serial.print(adc1);
  Serial.print("\t");
  Serial.print(adc2);
  Serial.print("\t");
   Serial.print(adc3);
  Serial.print("\t");
  Serial.print(adc4);
  Serial.print("\t");
  Serial.println(yk);
  TC_GetStatus(TC1, 0);
}
```

*oczywiście zmienne yk,ukp muszą być wcześniej zadeklarowane i zainicjalizowane!!!

Co możemy pobadać

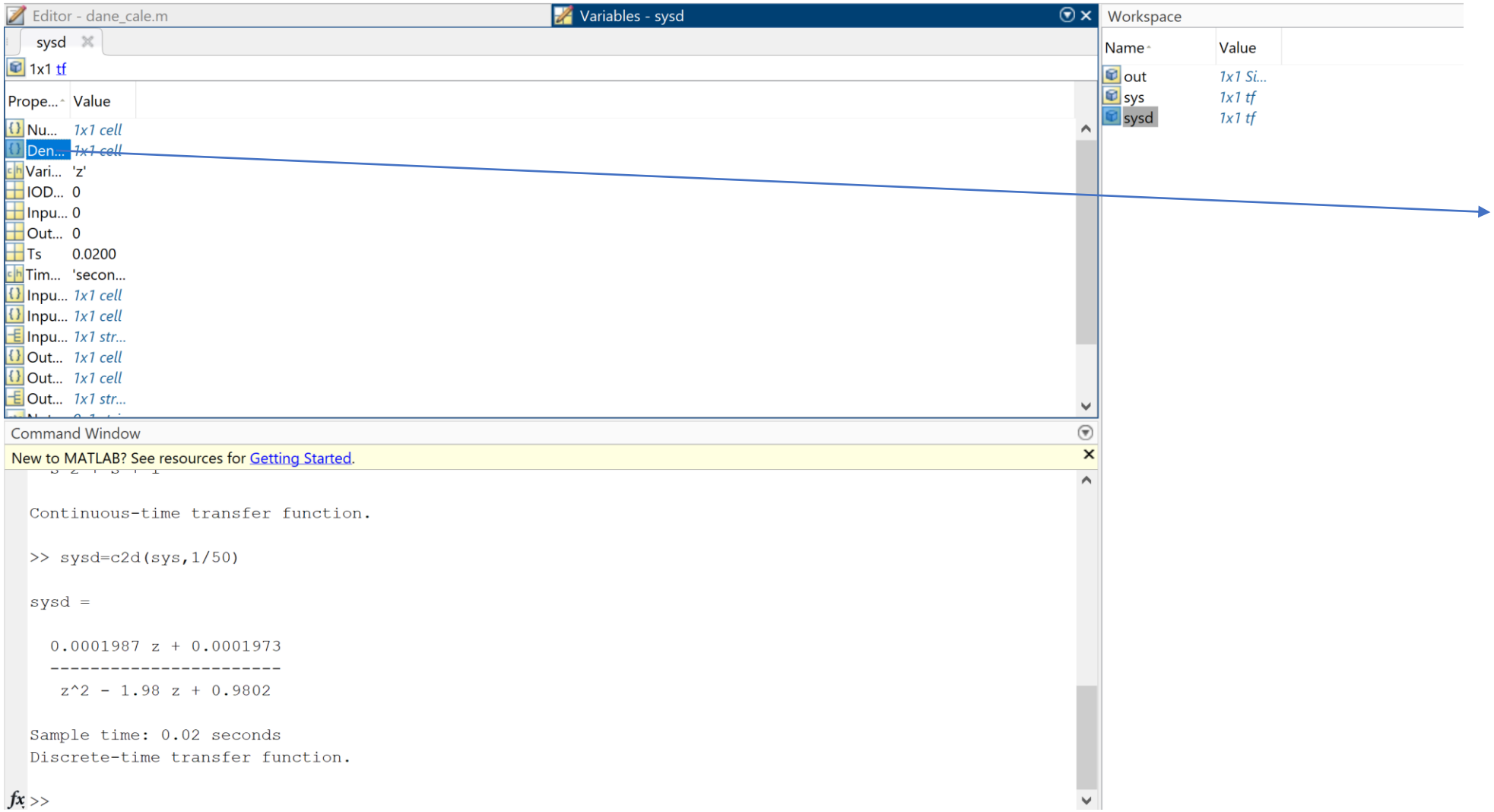
1. Porównanie odpowiedzi układu ciągłego i dyskretnego dla różnych częstotliwości sygnału (także przekraczających granicę Shanona)
2. Zaimplementowanie dyskretnego odpowiednika układu ciągłego o innych parametrach np. 1/(s+2)
3. Zachowanie układu przy błędnie określonym okresie próbkowania
   (#define fs 50
      #define Ts 1/fs)
4. Przebadanie zachowania układu, gdy umieścimy kod wyznaczający dyskretną odpowiedź w funkcji loop()

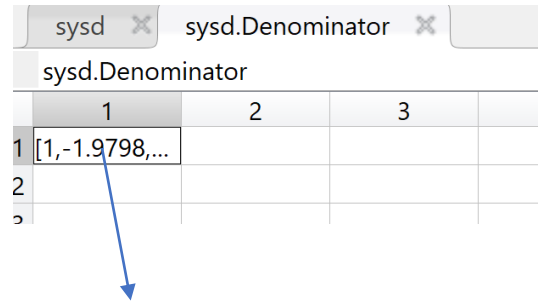
Implementacja układu drugiego rzędu

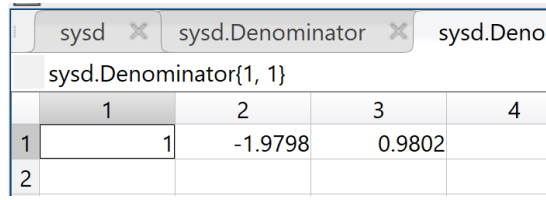
```
>> sys=tf([1],[1 1 1])

sys =

      1
  -----------
  s^2 + s + 1

Continuous-time transfer function.
```

```
>> sysd=c2d(sys,1/50)

sysd =

  0.0001987 z + 0.0001973
  -----------------------
   z^2 - 1.98 z + 0.9802

Sample time: 0.02 seconds
Discrete-time transfer function.
```

Problem dokładności współczynników







Co możemy pobadać

1. Wpływ dokładności współczynników na odpowiedź układu
2. Zaimplementowanie dyskretnego odpowiednika układu ciągłego o innych parametrach