

ANALIZA SYGNAŁÓW

ZUZANNA NASIŁOWSKA

GENEROWANIE SYGNAŁU	<input type="checkbox"/>	21.03.2024 r.
	<input type="checkbox"/>	nr. 276023
	<input type="checkbox"/>	II ROK, MST
	<input type="checkbox"/>	dr inż. Ireneusz Augustyniak

1 Wstęp

Zrealizowane ćwiczenie polegało na wygenerowaniu sygnału w programie Scilab, przy pomocy narzędzi graficznego edytora Xcos. Konkretnie wykonywaliśmy kontur znaku D-42 (obszar zabudowany), który uległ lekkiej modyfikacji, a dokładniej odbiciu względem osi OX oraz OY. Celem ćwiczenia jest zapoznanie się z wyżej wymienionym programem, a także wykonanie wspomnianego sygnału, który możemy zobaczyć poniżej:



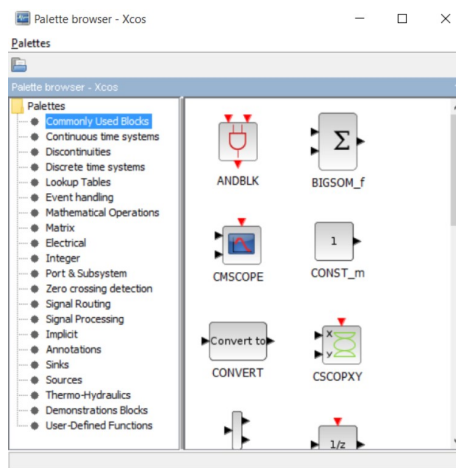
Rysunek 1: Zadany sygnał

W niniejszym raporcie opracowane zostaną kroki realizacji zadanego sygnału wraz z omówieniem użytych funkcji.

2 Realizacja ćwiczenia

2.1 Opis użytych bloków

Przed wygenerowaniem konkretnego sygnału należy zapoznać się z dostępnymi w programie **blokami**, które znajdziemy w katalogu "Palettes" [po uruchomieniu Xcos jest to jedno z domyślnie otwartych okien]:

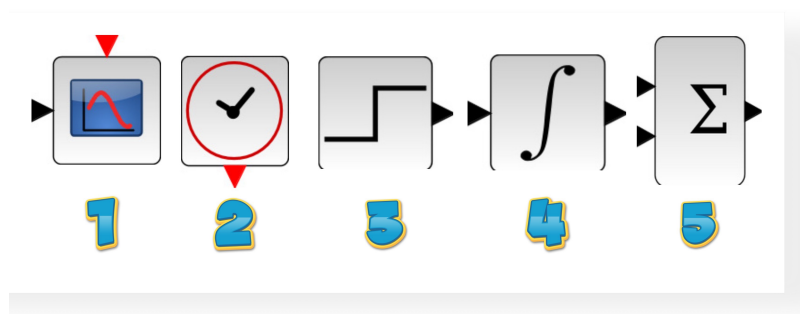


Bloki posiadają przeróżne właściwości umożliwiające modelowanie wskazanych sygnałów, znajomość ich działania pomoże nam w otrzymaniu m.in. oczekiwanych w zadaniu skosów.

W realizowanym ćwiczeniu wykorzystaliśmy dokładnie **pięć bloków**:

- **CSCOPE** - oscyloskop, dostosowuje ilość czasu i zakres wyświetlanych danych wejściowych oraz wyświetla je w odniesieniu do czasu symulacji. Dodatkowo posiada 10 parametrów, którymi możemy operować.
- **CLOCK_c** - zegar, którego wyjście generuje regularny ciąg zdarzeń, zgodnie z ustalonym okresem przy pomocy określenia parametru **Period**.
- **STEP_FUNCTION** - funkcja skoku jednostkowego, blok skokowy, który wysyła sygnał krokowy między dwoma poziomami (wartość początkowa **Initial Value** i wartość końcowa **Final Value**, rozpoczynając od ustalonego czasu kroku **Step Time**).

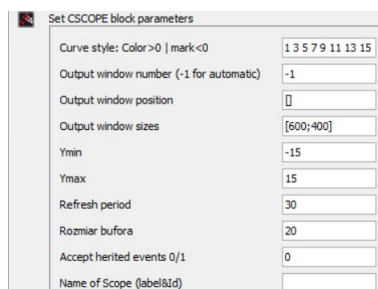
- **INTEGRAL_m** - blok całkujący, wartość wyjściowa jest wynikiem całkowania w danym kroku czasowym.
- **BIGSOM_f** - blok sumujący, wzmacnia lub osłabia sygnał o dane wejściowe. Użytkownik może dostosować wzmacnienie dla każdego wejścia za pomocą parametru **Inputs ports signs/gain**.



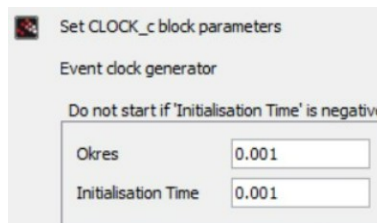
2.2 Tworzenie sygnału

Projektowanie naszego sygnału zapoczątkujemy ustaleniem parametrów dla bloków **1**, **2** oraz **5**.

- Blok **CSCOPE** zastosujemy jeden raz wraz z domyślnymi parametrami prezentującymi się następująco:



- Blok **CLOCK_c** użyjemy również tylko jeden raz i zmienimy parametry w taki sposób, aby wyprostować linie pionowe naszego sygnału, dokładniej zmienimy parametry domyślne z wartości 0,1 na 0,001:
- Blok **BIGSOM_f** użyjemy 6 razy z czego jeden blok będzie głównym od którego odgałęzimy pozostałe pięć. Parametry w naszym bloku będziemy dostosowywać w zależności od liczby połączeń z naszym blokiem:

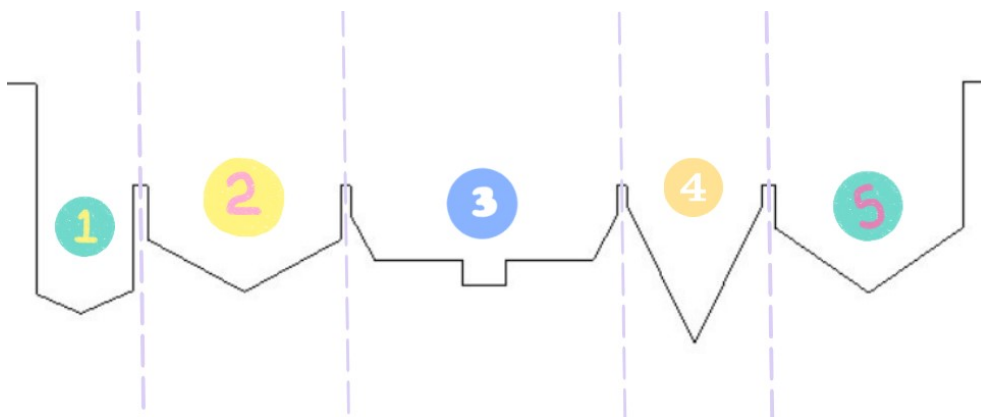


Rysunek 2: Parametry zegara



Rysunek 3: Przykładowe zastosowane wartości parametru [szary kolor bloku na schemacie]

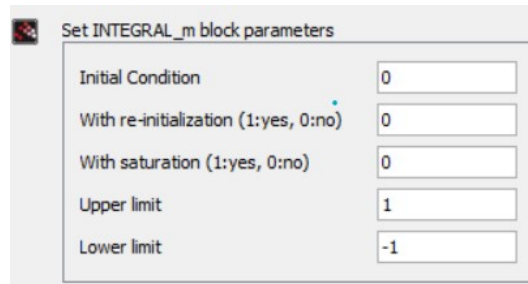
W następnym kroku tworzenia podzielimy nasz sygnał na **5 części**, aby całość była dla nas bardziej czytelna. Poniżej przedstawiony został pomocniczy rysunek wraz z numeracją kolejności wykonywanych budynków:



Rysunek 4: **Kolejność wykonywania budynków sygnału**

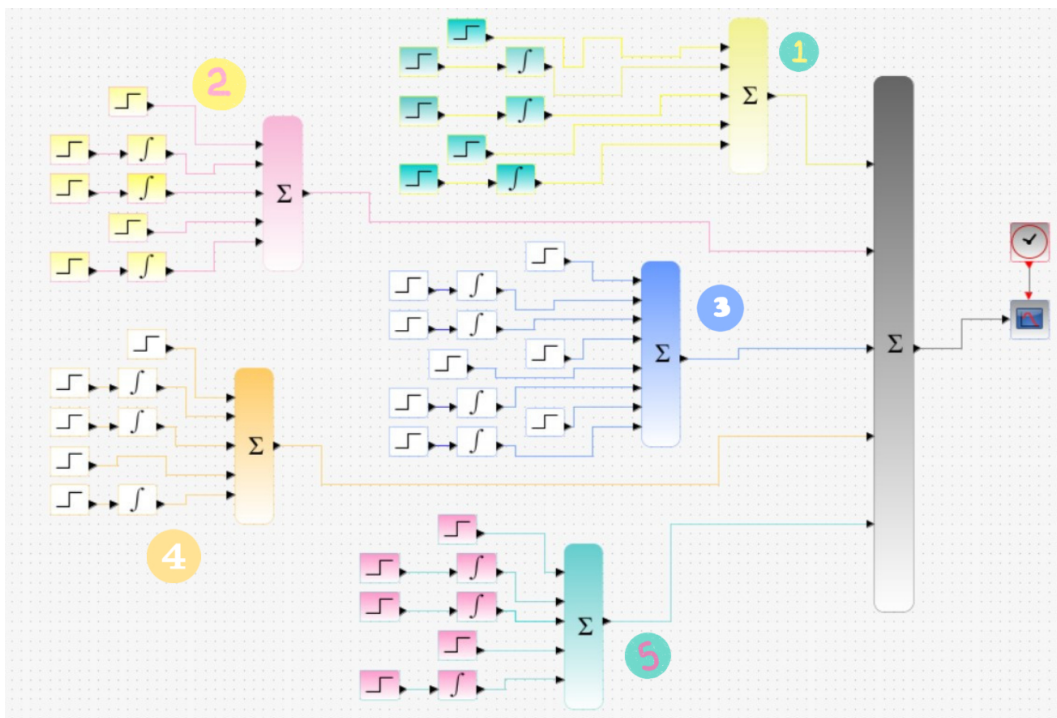
W każdej części zajęliśmy się tworzeniem wskazanego budynku. W tym celu wykorzystaliśmy bloki numer **3**, **4** oraz **5**.

Parametry **bloku całkującego** zostawiliśmy domyślne bez zmian w każdym przypadku, gdzie blok został zastosowany (w wypadku, gdzie nie został użyty oznaczmy parametry poprzez X w później przedstawionych tabelkach):



2.3 Główny schemat sygnału oraz pozostałe parametry

W następnej kolejności możemy przedstawić zbudowany schemat, na którym widzimy pojedyncze sekcje numerycznie odpowiadające naszej kolejności wykonywania budynków:



Rysunek 5: Schemat blokowy

Każda taka sekcja wykonuje sygnał pojedynczej budowli znaku, połączenie zawierające sam blok *STEP_FUNCTION*, będzie w rezultacie tworzyć pion, zaś połączenie składające się z bloków *STEP_FUNCTION* oraz *INTEGRAL_m* utworzy w naszym przypadku wszystkie skosy oraz niektóre

linie poziome.

Przyjrzyjmy się jeszcze na parametry bloku **STEP_FUNCTION**. Jak wspomniano wyżej mamy styczność z trzema parametrami: **Step Time**, **Initial Value**, **Final Value**. W naszym przypadku wartość Initial Value, będzie zawsze równa 0.



Rysunek 6: Przykładowe parametry

Parametry





Przedstawione poniżej tabelki odpowiadają kolorystycznie oraz numerycznie każdej sekcji widocznej na schemacie. Numerki w tabelce [1,2...] odpowiadają blokom w pojedynczej sekcji zaczynając od góry, więc wszystko czytamy od góry do dołu [zaś parametry czytamy odpowiednio poziomo].

CZĘŚĆ 1						
L.P.	BLOK 1	Step Time	Initial Value	Final Value	BLOK 2	PARAMETRY
1	STEP_FUNCTION	1	0	-5	INTEGRAL_m	
2	STEP_FUNCTION	1	0	-1	INTEGRAL_m	domyślne
3	STEP_FUNCTION	1.46	0	2	INTEGRAL_m	domyślne
4	STEP_FUNCTION	2.0	0	2.5	INTEGRAL_m	
5	STEP_FUNCTION	2.0	0	-1	INTEGRAL_m	domyślne



Rysunek 7: Składniki i parametry części 1

CZĘŚĆ 2						
L.P.	BLOK 1	Step Time	Initial Value	Final Value	BLOK 2	PARAMETRY
1	STEP_FUNCTION	2.15	0	-1.28	INTEGRAL_m	
2	STEP_FUNCTION	2.15	0	-1.25	INTEGRAL_m	domyślne
3	STEP_FUNCTION	3.15	0	2.5	INTEGRAL_m	domyślne
4	STEP_FUNCTION	4.15	0	1.28	INTEGRAL_m	
5	STEP_FUNCTION	4.15	0	-1.25	INTEGRAL_m	domyślne



Rysunek 8: Składniki i parametry części 2

CZĘŚĆ 3						
L.P.	BLOK 1	Step Time	Initial Value	Final Value	BLOK 2	PARAMETRY
1	STEP_FUNCTION	4.245	0	-0.69	INTEGRAL_m	
2	STEP_FUNCTION	4.245	0	-4.3	INTEGRAL_m	domyślne
3	STEP_FUNCTION	4.50	0	4.3	INTEGRAL_m	domyślne
4	STEP_FUNCTION	5.40	0	-0.59	INTEGRAL_m	
5	STEP_FUNCTION	5.85	0	0.59	INTEGRAL_m	
6	STEP_FUNCTION	6.75	0	4.3	INTEGRAL_m	domyślne
7	STEP_FUNCTION	7.005	0	0.69	INTEGRAL_m	
8	STEP_FUNCTION	7.005	0	-4.3	INTEGRAL_m	domyślne

Rysunek 9: Składniki i parametry części 3

CZĘŚĆ 4						
L.P.	BLOK 1	Step Time	Initial Value	Final Value	BLOK 2	PARAMETRY
1	STEP_FUNCTION	7.1	0	-0.46	INTEGRAL_m	
2	STEP_FUNCTION	7.1	0	-4.7	INTEGRAL_m	domyślne
3	STEP_FUNCTION	7.8	0	9.4	INTEGRAL_m	domyślne
4	STEP_FUNCTION	8.5	0	0.46	INTEGRAL_m	
5	STEP_FUNCTION	8.5	0	-4.7	INTEGRAL_m	domyślne

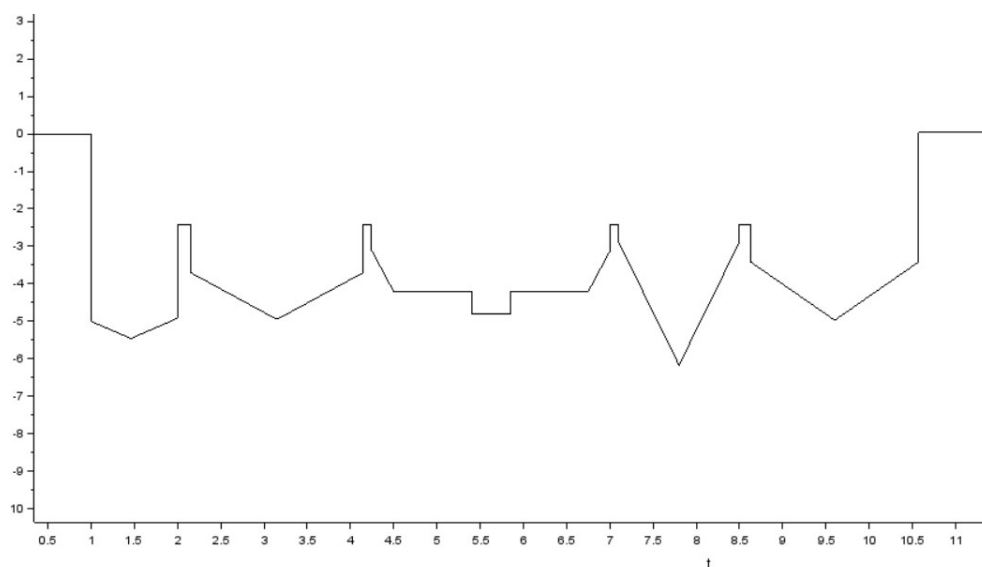
Rysunek 10: Składniki i parametry części 4

CZĘŚĆ 5						
L.P.	BLOK 1	Step Time	Initial Value	Final Value	BLOK 2	PARAMETRY
1	STEP_FUNCTION	8.63	0	-1	INTEGRAL_m	
2	STEP_FUNCTION	8.63	0	-1.6	INTEGRAL_m	domyślne
3	STEP_FUNCTION	9.6	0	3.2	INTEGRAL_m	domyślne
4	STEP_FUNCTION	10.57	0	3.455	INTEGRAL_m	
5	STEP_FUNCTION	10.57	0	-1.6	INTEGRAL_m	domyślne

Rysunek 11: Składniki i parametry części 5

2.4 Wygenerowany sygnał

Ostatecznie po wykonaniu wszystkich kroków otrzymujemy nasz kontur znaku obszaru zabudowanego:



Rysunek 12: Otrzymany sygnał

3 Wnioski

Przy pomocy narzędzi graficznego edytora Xcos możemy wygenerować konkretne sygnały przy zastosowaniu odpowiednich bloków. W naszym przypadku zaprojektowaliśmy sygnał przypominający kontur widoczny na znaku D-42. Podział pracy na poszczególne sekcje, odpowiadające konkretnym budynkom, z pewnością ułatwił analizę, a także pracę nad sygnałem. Największą trudnością okazały się skosy i precyzyjność ich wykonania, aby ostateczny kontur był jak najbardziej podobny do zadanego. Okazuje się, że dużym ułatwieniem są nasze osie wraz z ich podziałką, ponieważ możemy nasz zadany obrazek dopasować z naszymi osiami w innych programach. Odczytać wartości i dokładnie obliczyć dane, aby skosy były pod odpowiednim kątem lub odległości między ścianami budynków takie same.