FOURIEROV REZONATOR

Grubeša, Marko

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Sveučilišni studij

FOURIEROV REZONATOR

Završni rad

Marko Grubeša

Osijek, 2019.

SADRŽAJ

1. UVOD

Fourierova analiza je prisutna u raznim područjima znanosti, matematike, fizike, inženjerstva. Fourierova analiza koristi se za rješavanje realnih i važnih problema. Jako bitnu ulogu ima u obradi signala kao što su zvuk, valovi i slike. Fourierova analiza nastala je istraživanjem Fourierovog reda nazavnog po francuskom matematičaru Jean-Baptiste Joseph Fourier. Fourier se bavio problemima prijenosa topline i vibracijama, te je pokazao da se rastavljanjem složene funkcije na jednostavne sume trigonometrijskih funkcija pojednostavljuju ti problemi. Taj proces dekompozicije složene funkcije na jednostavne komponente je upravo ono čime se bavi Fourierova analiza. Sam proces dekompozicije naziva se Fourierova transformacija. Koriste se razni oblici Fourierove transformacije.

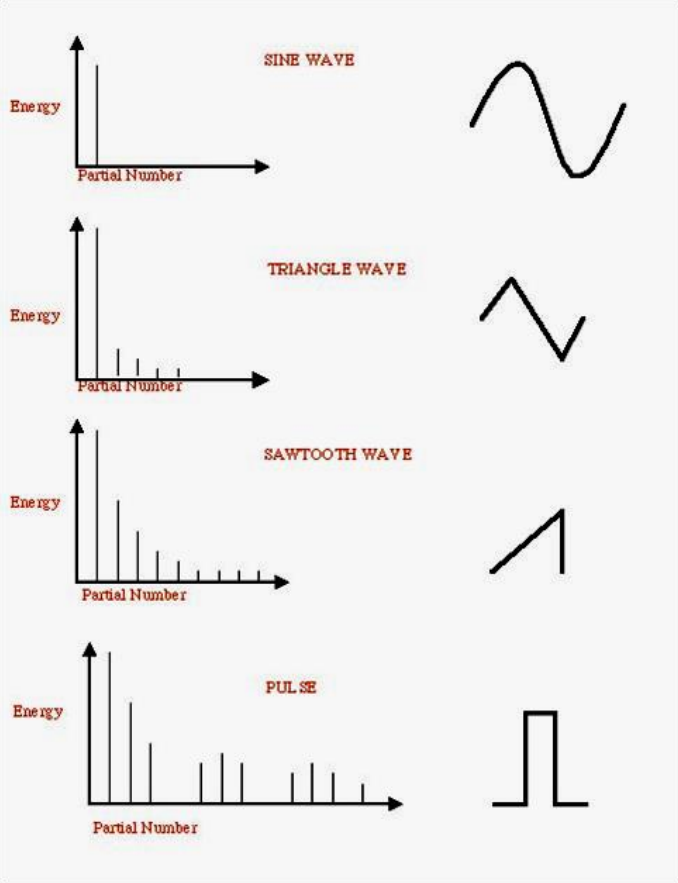
Ovaj se rad sastoji od šest poglavlja. U prvom poglavlju dan je uvod i opis zadatka. U drugom poglavlju opisana je Fourierova analiza. U trećem poglavlju dan je opis alata koji su se koristili za izradu programa i opis programabilnog logičkog kontrolera(engl. *Programmable Logic Controller* - PLC). U četvrtom poglavlju opisano je programsko rješenje i vizualizacija. U petom poglavlju je sažetak rada.

1.1. Zadatak završnog rada

U ovom završnom radu zadatak je izraditi programski blok za Fourierovu analizi ulaznog signala, tj. razložiti ulazni signal na sinusove i kosinusove komponente. Program treba razviti u alatu Step7 i Codesys te vizualizaciju prikazati u alatu Codesys.

2. FOURIEROVA ANALIZA

Jedan od principa Fourierove analize je da se svaki signal može konstruirati kao suma sinusnih valova, koji imaju različite amplitude, faze i frekvencije. Fourierova analiza svodi se na Fourierovu transformaciju.[1] Signal se može promatrati u vremenskoj ili frekvencijskom domeni. Prikaz signala u frekvencijskoj domeni naziva se spektar signala (Slika 2.1). Pomoću Fourierove transformacije vremenski kontinuirani signal može se prikazati kao suma više različitih frekvencijskih komponenti[2].



***Slika 2.1**: Spektar signala*[1]

Postoje četiri oblika Fourierove transformacije: vremenski kontinuirana Fourierova transformacija (engl. *Continuous Time Fourier Transform*-CTFT), razvoj u Fourierov red (engl. *Continuous Time Fourier Series*-CTFS), diskretna Fourierova transformacija (engl. *Discrete Fourier transform-*

[1] http://physics.mef.hr/Predavanja/java%20matem%20fje/fouriert.htm

DFT),Vremenski diskretna Fourierova transformacija (engl. *The Discrete Time Fourier Transform*-DTFT).

2.1. Vremenski kontinuirana Fourierova transformacija (CTFT)

Fourierova transformacija je reverzibilna, linearna transformacija s mnogo svojstava. Za svaku funkciju x(t) Fourierova transformacija može biti označena s X(iω), gdje je X(iω) složena funkcija frekvencije ω[3]. Često t predstavlja vremensku domenu, a iω frekvencijsku domenu. Fourierova transformacija definirana je sljedećim izrazom:

$$X(i\omega) = \int_{-\infty}^{\infty} x(t)e^{-i\omega t}dt \tag{2-1}$$

Njena inverzna transformacija glasi:

$$x(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} X(i\omega)e^{i\omega t}d\omega \tag{2-2}$$

Vremenski kontinuirana Fourierova transformacija definira frekvencijsku domenu, gdje je frekvencijska domena neprekidna i neograničena.

Dovoljni (ali ne i nužni) uvjeti za postojanje Fourierove transformacije funkcije x(t) su:

1.Funkcija x(t) zadovoljava Dirichletove uvjete (funkcija je ograničena s konačnim brojem maksimuma i minimuma te konačnim brojem diskontinuiteta u bilo kojem konačnom vremenskom intervalu).[4]

2. $\int_{-\infty}^{+\infty}|x(t)|dt < \infty$ (2-3)

Tablica 2.1. *Svojstva Fourierove transformacije*

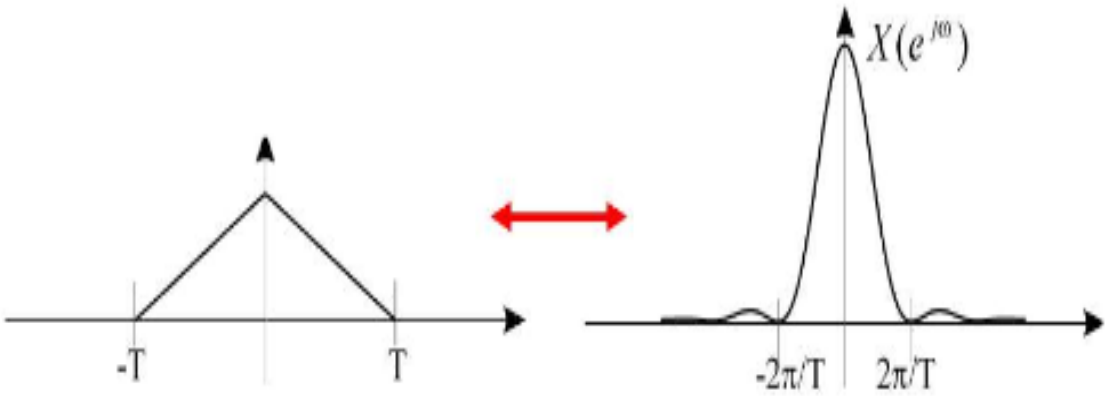
Svojstva	*Vremenska domena*	*Frekvencijska domena*

Linearnost	$ax_1(t) + bx_2(t)$	$aX_1(i\omega) + bX_2(i\omega)$
Vremenska inverzija	$x(-t)$	$X(-i\omega)$
Vremenski pomak	$x(t+\tau)$	$X(i\omega)e^{i\omega\tau}$
Frekvencijski pomak	$x(t)e^{im\omega t}$	$X(i\omega + m)$
Konvolucija u vremenskoj domeni	$f(t) * g(t)$	$F(i\omega)G(i\omega)$
Konvolucija u frekvencijskoj domeni	$f(t)g(t)$	$F(i\omega) * G(i\omega)$
Parsevalova jednakost	$P = \int_{-\infty}^{\infty} \lvert x(t)\rvert^2 dt$	$P = \frac{1}{2\pi}\int_{-\infty}^{\infty} \lvert X(i\omega)\rvert^2 d\omega$

Na slici 2.2 vidi se primjer transformacije trokutnog signala[5]:

$$x(t) = \begin{cases} t - \left|\frac{t}{T}\right|, za\ |t| \le T \\ 0, za\ |t| > T \end{cases} \quad (2\text{-}4)$$

$$TR(\overline{\omega}) = Tsinc^2(\overline{\omega}T/2) \quad (2\text{-}5)$$



***Slika 2.2**:Transformacija trokutnog signala*[2]

[2] https://loomen.carnet.hr/pluginfile.php/2163840/mod_resource/content/1/Predavanje%2014.pdf

2.2. Fourierov red(CTFS)

Uobičajeni je postupak u matematici da se složenije funkcije prikazuju pomoću jednostavnijih. Na primjer, eksponencijalnu funkciju može se prikazati kao zbroj reda[5]:

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots + \frac{x^n}{n!} + \cdots \tag{2-6}$$

Tako se može aproksimirati eksponencijalna funkcija polinomom, uzimajući konačno mnogo pribrojnika s desne strane. Taylorovim redom mogu se prikazati samo funkcije koje zadovoljavaju stroge uvjete. Na primjer, prekinuta se funkcija ne može prikazati Taylorovim redom. Taylorov red glasi[6]:

$$f(x) = f(x_0) + \sum_{n=1}^{\infty} \frac{f^{(n)}(x_0)}{n!}(x - x_0)^n \tag{2-7}$$

Međutim, umjesto potencija 1, x, x^2, ….. može se odabrati neki drugi pogodni skup funkcija kojim će prikazivati složenije funkcije. Joseph Fourier utvrdio je da se svaka funkcija na ograničenom intervalu može prikazati kao suma različitih amplituda, faza i frekvencija. Vremenski kontinuirani periodički signal s periodom T[5]:

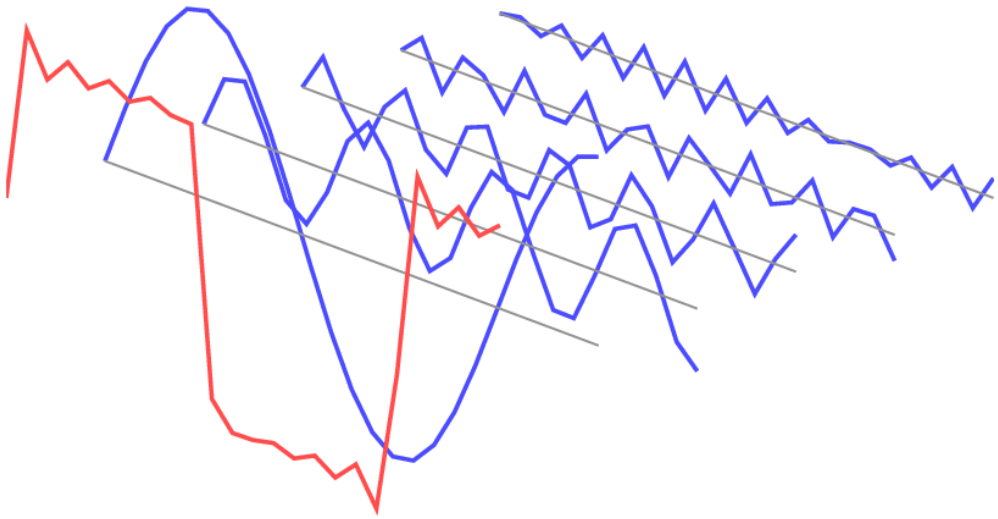
$$\tilde{x}(t) = \tilde{x}(t + nT), \quad n \in \mathbb{Z} \tag{2-8}$$

se može razviti u red:

$$\tilde{x}(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \sin(\omega n t + \varphi_n), \quad n \in \mathbb{Z}, \omega = \frac{2\pi}{T} \tag{2-9}$$

Fourierov red se može zapisati i ovako:

$$\tilde{x}(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos(\omega n t) + \sum_{n=1}^{\infty} b_n \sin(\omega n t), n \in \mathbb{Z} \quad (2\text{-}10)$$



***Slika 2.3**:Primjer rastava signala na komponente*[3]

Na slici 2.3 prikazan je 3-D rastav signala na komponente.

Fourierovi koeficijenti a_0, a_n, b_n računaju se na slijedeći način[7]:

$$a_0 = \frac{1}{T}\int_{-T/2}^{T/2} \tilde{x}(t)dt \quad (2\text{-}11)$$

$$a_n = \frac{2}{T}\int_{-T/2}^{T/2} \tilde{x}(t)\cos(\omega n t)dt \quad (2\text{-}12)$$

$$b_n = \frac{2}{T}\int_{-T/2}^{T/2} \tilde{x}(t)\sin(\omega n t)dt \quad (2\text{-}13)$$

[3] https://i.stack.imgur.com/Mnq4i.png

Pomoću formule (2-10) može se primjenom Eulerove formule:

$$e^{i\varphi} = cos\varphi + isin\varphi \tag{2-14}$$

Fourierov red se može zapisati na slijedeći način:

$$\tilde{x}(t) = \frac{a_0}{2} + \sum_{n=-\infty}^{\infty} X(n)e^{i\omega nt}, \ \ n \in \mathbb{Z} \tag{2-15}$$

Fourierovi koeficijenti onda glase:

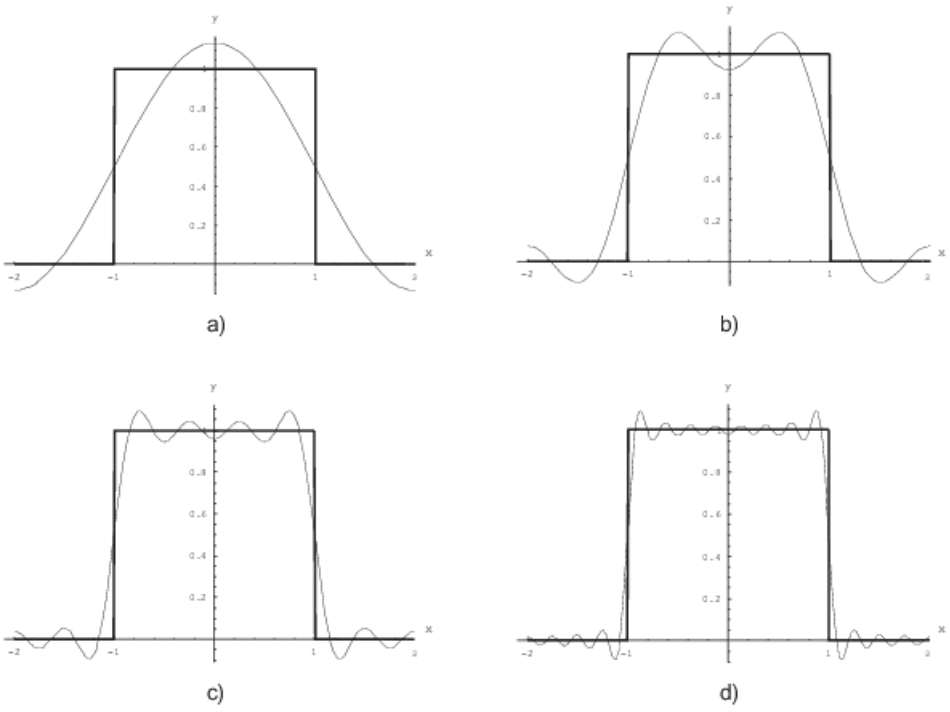
$$X(n) = \frac{1}{T}\int_{-T/2}^{T/2} \tilde{x}(t)e^{-i\omega nt}dt \tag{2-16}$$

Tablica 2.2. *Svojstva Fourierovog reda*

Svojstva	*Vremenska domena*	*Frekvencijska domena*
Linearnost	$a\tilde{x}_1(k) + b\tilde{x}_2(k)$	$a\tilde{X}_1(n) + b\tilde{X}_2(n)$
Vremenska inverzija	$\tilde{x}(-k)$	$\tilde{X}(-n)$
Vremenski pomak	$\tilde{x}(k+\tau)$	$\tilde{X}(n)e^{i\omega n\tau}$
Frekvencijski pomak	$\tilde{x}(k)e^{im\omega k}$	$\tilde{X}(n+m)$
Konvolucija u vremenskoj domeni	$\tilde{f}(k) * \tilde{g}(k)$	$\tilde{F}(n)\tilde{G}(n)$
Konvolucija u frekvencijskoj domeni	$\tilde{f}(k)\tilde{g}(k)$	$\tilde{F}(n) * \tilde{G}(n)$
Parsevalova jednakost	$P = \frac{1}{N}\sum_{k=0}^{N-1}\lvert\tilde{x}(k)\rvert^2$	$P = \sum_{n=0}^{N-1}\lvert\tilde{X}(n)\rvert^2$

Na slici 2.4 prikazan je primjer Fourierovog reda na pravokutnoj funkciji[8]:

$$x(t) = \begin{cases} 1, & t \in (-1,1) \\ 0, & inače \end{cases} \tag{2-17}$$



***Slika 2.4**:Fourierov red za pravokutni signal a)n=2 b)n=4 c)n=8 d)n=16*[4]

2.3. Diskretna Fourierova transformacija (DFT)

Diskretna Fourierova transformacija je ekvivalent vremenski kontinuirane Fourierove transformacije. DFT pretvara konačni niz jednako razmaknutih uzoraka funkcije u istu duljinu slijeda jednako razmaknutih uzoraka vremenski diskretne Fourierove transformacije (DTFT). Dakle, ulazni signal u DFT-u je izlazni signal u DTFT-u. DFT je najvažnija diskretna transformacija koja se koristi za izvođenje Fourierove analize. Koristi se za numeričko određivanje spektra signala. U digitalnoj obradi signala, funkcija je bilo koji signal koji se mijenja tijekom vremena, kao što je radio signal. Signal je uzorkovan tijekom konačnog vremenskog intervala. DFT se također koristi za učinkovito rješavanje parcijalnih diferencijalnih jednadžbi i za

[4] http://e.math.hr/math_e_article/br19/matijevic

obavljanje drugih operacija kao što su konvolucije ili množenje velikih cijelih brojeva.[10] Pošto se bavi konačnim nizom podataka, može se implementirati na različite načine. Te implementacije često koriste algoritme brze Fourierove transformacije (FFT). DFT može se zapisati na slijedeći način[11]:

$$X_k = \sum_{n=0}^{N-1} x[n] e^{\frac{-jkn2\pi}{N}} \tag{2-19}$$

De Moiverov teorem govori da je $e^{j\theta} = \cos(\theta) + jsin(\theta)$ (2-19), te onda DFT glasi:

$$X_k = X_{re}[k] + jX_{im}[k] \tag{2-20}$$

Gdje su $X_{re}[k]$ i $X_{im}[k]$ jednaki:

$$X_{re}[k] = \sum_{n=0}^{N-1} x[n] \cos\left(\frac{kn2\pi}{N}\right) \tag{2-21}$$

$$X_{im}[k] = -\sum_{n=0}^{N-1} x[n] \sin\left(\frac{kn2\pi}{N}\right) \tag{2-22}$$

Inverzni DFT glasi:

$$x[n] = \frac{1}{N} \sum_{n=0}^{N-1} X[k] e^{\frac{jkn2\pi}{N}} \tag{2-23}$$

Najveća frekvencijska komponenta spektra dobivenom DFT-om naziva se Nyquistova frekvencija [13]:

$$f_{max} = \frac{f_s}{2} \tag{2-24}$$

f_s je frekvencija kojom je signal otipkan. Može doći do Alias efekta ako signal sadrži komponente veće od Nyquistove frekvencije. Alias efekt je posljedica nedovoljno gustog uzorkovanja signala. Izbjegava se tako da se signali filtriraju prije utipkavanja. Alias efekt se isto može izbjeći ako se poveća frekvencija utipkavanja, ali onda može doći do smanjenja rezolucije spektra[13].

Za uzorke DFT-A moguće je izračunati amplitudu |X[k]| i fazu ∠X[k] koje predstavljaju otipkanu amplitudnu i faznu karakteristiku (spektar) prema izrazima:

$$|X[k]| = \sqrt{Re\{X[k]\}^2 + Im\{X[k]\}^2} \tag{2-25}$$

$$\angle \mathrm{X[k]} = \tan^{-1} 2(Im\{X[k]\}, Re\{X[k]\}) \tag{2-26}$$

Tablica 2.3. *Svojstva DFT-a*

Svojstva	*Vremenska domena*	*Frekvencijska domena*
Linearnost	$ax[n] + by[n]$	$aX[k] + bY[k]$
Vremenska inverzija	X[(-n) mod N]	X[(-k) mod N]
Konjugirani par	$x^*[n]$	$X^*[(-k) \bmod N]$
Kružna konvolucija	$x[n] * y[n]$	$X[k]Y[k]$
Kružna korelacija	$x[n] * y^*[-n]$	$X[k]Y^*[k]$
Množenje	$x[n]y[n]$	$\frac{1}{N}X[k] * Y[k]$
Parsevalova jednakost	$P = \sum_{k=0}^{N-1} x[n] * y^*[n]$	$P = \frac{1}{N}\sum_{k=0}^{N-1} X[k] * Y^*[k]$

	$P = \sum_{k=0}^{N-1} \lvert x[n] \rvert^2$	$P = \frac{1}{N} \sum_{k=0}^{N-1} \lvert X[k] \rvert^2$

2.4. Vremenski diskretna Fourierova transformacija (DTFT)

DTFT se koristi za analizu uzoraka kontinuirane funkcije. Transformacija se vrši nad uzorkovanim signalima duljine N. Signali su definirani na području cijelih brojeva. Ona gradi neograničeni kontinuirani signal [14]. Povezana je s diskretnom Fourierovom transformacijom (DFT). DTFT glasi:

$$X(\omega) = \sum_{n=-\infty}^{+\infty} x(n) e^{-j\omega n} \tag{2-27}$$

Njena inverzna transformacija glasi:

$$x(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(w)\, e^{j\omega n} d\omega \tag{2-28}$$

Tablica 2.4. *DTFT-a*

Svojstva	*Vremenska domena*	*Frekvencijska domena*
Linearnost	$ax[n] + bg[n]$	$aX(e^{j\omega}) + bG(e^{j\omega})$
Vremenski pomak	$e^{jyn}x[n]$	$e^{-j\omega n_0}X(e^{j\omega})$
Diferencija u frekvenciji	$-jnx[n]$	$X(e^{j(\omega-y)}$
Konvolucija-Množenje	$-jnx[n]$	$\frac{d}{d\omega}X(e^{j\omega})$
Množenje-Konvolucija	$-jnx[n]$	$X(e^{j\omega})G(e^{j\omega})$
Množenje-Konvolucija	$-jnx[n]$	$\frac{1}{2\pi}\int_{-\pi}^{\pi} X(e^{jy})G(e^{j(\omega-y)})dy$
Parsevalova jednakost	$P = \sum_{n=-\infty}^{\infty} \lvert x[n]\rvert^2$	$P = \frac{1}{2\pi}\int_{-\pi}^{\pi} \lvert X(e^{j\omega})\rvert^2 \, d\omega$

2.5. Brza Fourierova transformacija (FFT)

Brza Fourierova transformacija je brz način izračuna diskretne Fourierove transformacije (DFT). Korištenjem periodičnosti u sinusnim signalima, FFT smanjuje vrijeme izračuna transformacije. FFT može ubrzati račun ako je broj uzoraka koji se transformira jednak potenciji broja 2, tako se mogu eliminirati nepotrebne operacije. Kao rezultat toga smanjuje složenost računanja DFT-a s $O(n^2)$ na O(nlogn), gdje je n duljina. Razlika u brzini može biti velika, posebno za velike nizove podataka. Postoje razni FFT algoritmi od kojih je najpopularniji takozvani Cooley–Tukey algoritam.[11]

2.5.1. Cooley-Turkey algoritam

Cooley-Turkey algoritam funkcionira tako da dijeli niz podataka koji će se transformirati u manje nizove podataka. Potom, dijeli te manje nizove podataka u još manje nizove podataka. U svakoj fazi obrađeni rezultati prethodne faze se kombiniraju na poseban način. Konačno izračunava se DFT svakog malog niza podataka[11]. Na primjer, FFT niza od 32 podataka se dijeli na 2 FFT-a od 16 podataka, on se potom dijeli na 4 FFT-a od 8 podataka, a oni se dijele na 8 FFT-ova od 4 podataka, te ćemo na kraju imati 16 FFT-ova, pri čemu se svaki FFT sastoji od 2 podatka. DFT se raspisuje onda na slijedeći način[13]:

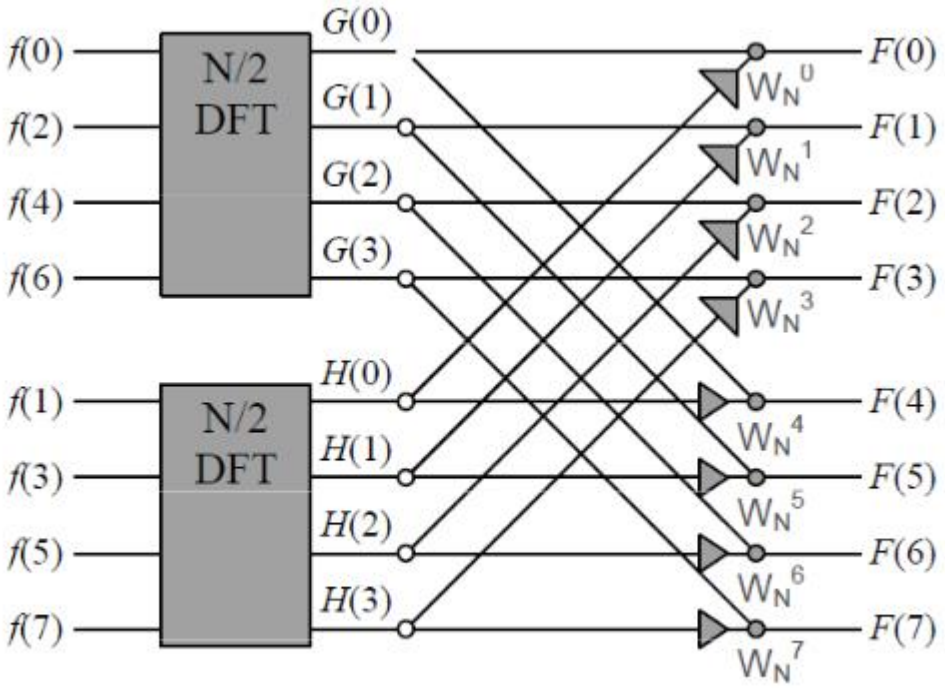
$$\begin{aligned} X_k &= \sum_{n=0}^{N-1} x[n] e^{\frac{-jkn2\pi}{N}} = \sum_{0}^{\frac{N}{2}-1} x[2n] e^{\frac{-jk2n2\pi}{N}} + \sum_{0}^{\frac{N}{2}-1} x[2n+1] e^{\frac{-jk(2n+1)2\pi}{N}} \\ &= \sum_{0}^{\frac{N}{2}-1} x[2n] e^{\frac{-jkn2\pi}{\frac{N}{2}}} + e^{\frac{-j2\pi k}{N}} \sum_{0}^{\frac{N}{2}-1} x[2n+1] e^{\frac{-jkn2\pi}{\frac{N}{2}}} \\ &= DFT\ \frac{n}{2}[x(0) + x(2) + x(4) + \cdots + x(N-2)] \\ &+ W(kn) DFT \frac{n}{2}[(x(1) + x(3) + \cdots + \cdots + x(N-1)] \end{aligned} \tag{2-29}$$

W(kn) je twiddle faktor i jednak je:

$$W(kn) = e^{\frac{-j2\pi k}{N}} \tag{2-30}$$

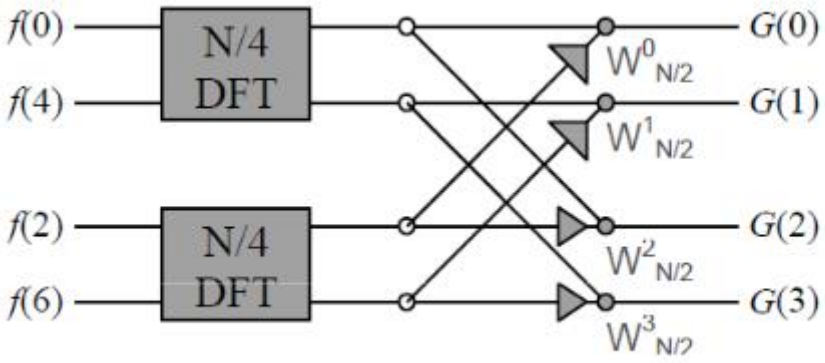
Dobila su se dva manja DFT-a od kojih jedan sadrži parne koeficijente, a drugi sadrži neparne koeficijente. Taj postupak razlaganja koeficijenata nazivamo decimacija u vremenu[12].

Na slici 2.5 korištenjem Cooly-Turkey algoritma dobivaju se dva manja FFT-a.



***Slika 2.5**:Rastav na dva manja FFT-a*[5]

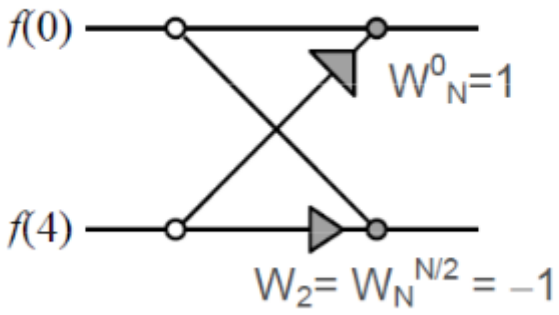
Ako je $\frac{N}{2}$ i dalje paran broj onda se može iz $\frac{N}{2}$ DFT odrediti s dva DFT s $\frac{N}{4}$ uzoraka kao što je prikazano na slici 2.6



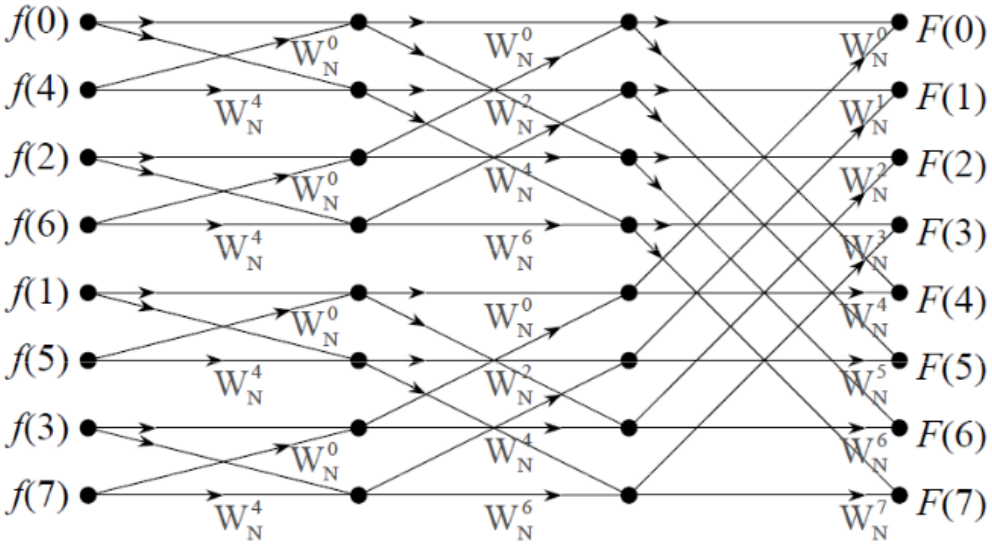
***Slika 2.6**:Daljnji Rastav na manje FFT-ove*[6]

[5] https://loomen.carnet.hr/pluginfile.php/281092/mod_resource/content/1/Stara_Predavanja/SiS15.pdf
[6] https://loomen.carnet.hr/pluginfile.php/281092/mod_resource/content/1/Stara_Predavanja/SiS15.pdf

Na kraju se dobije FFT od dva uzoraka ili DFT leptir (Slika 2.7) :



***Slika 2.7**:DFT leptir*[7]



***Slika 2.8**:Blok dijagram za N=8*[8]

Na slici 2.8 prikazan je postupak algoritma putem blok dijagrama[12].

[7] https://loomen.carnet.hr/pluginfile.php/281092/mod_resource/content/1/Stara_Predavanja/SiS15.pdf
[8] https://loomen.carnet.hr/pluginfile.php/281092/mod_resource/content/1/Stara_Predavanja/SiS15.pdf

3. KORIŠTENI ALATI

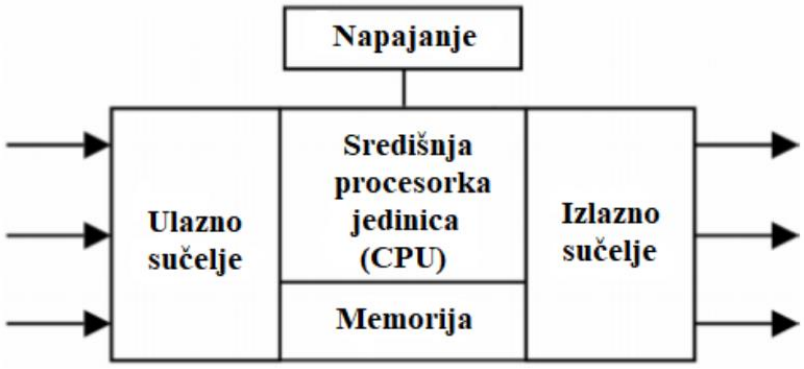
3.1. Programabilni logčki kontroler

PLC je programibilni logički kontroler, tj. industrijsko računalo koje je izumljeno prvenstveno kako bi zamijenilo postojeće sekvencijalne relejne krugove u upravljanju proizvodnim pogonima u industriji (Slika 3.1). Prednosti PLC-a su smanjenje dimenzija i potrošnje električne energije, pouzdanost i programibilnost, što im omogućuje široku primjenu u različitim postrojenjima u industriji[15].

***Slika 3.1**:Siemens Simatic S7 300 PLC*[9]

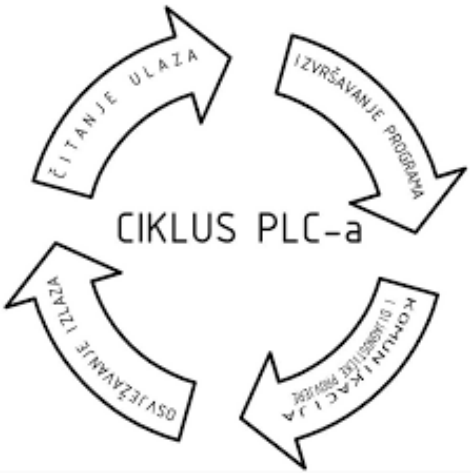
PLC uređaj sastoji se od centralne procesorske jedinice (CPU), ulaznih i izlaznih dijelova koji mogu biti digitalni ili analogni, memorijskog bloka, mrežnog modula koji služi za napajanje i komunikaciju te modula za proširenje. Analogni moduli mogu biti strujni ili naponski. Strujni analogni moduli upotrebljavaju se ako se signal mora slati na veće udaljenosti i raspon signala je od 4-20 miliampera. Naponski analogni moduli imaju raspon signala od 0-5 volti, 0-10 volti ili +/-10 volti. Digitalni moduli najčešće imaju 24 volta istosmjerne struje[15]. Funkcijske cjeline PLC-a prikazane su na slici 3.2.

[9] https://5.imimg.com/data5/GA/VV/MY-9815013/siemens-sinamics-s7-300-plc-500x500.jpg



***Slika 3.2**:Funkcijske cjeline PLC-a*[10]

PLC funkcionira tako da se program izvršava ciklički (Slika 3.3)



***Slika 3.3**:Izvršavanje programa PLC-a*[11]

3.2. Step7

Step7 je standardni softverski paket kojeg je razvila tvrtka Siemens. Koristi se za konfiguriranje i programiranje PLC-ova. Zasnovan je na IEC 61131-3 standardu. Podržava sljedeće programske jezike[16]:

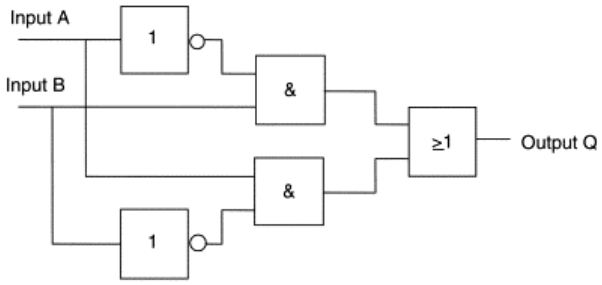
- FBD (engl. *Function Block Diagram*)
- LAD (engl. *Ladder Logic*)

[10] https://repozitorij.etfos.hr/islandora/object/etfos%3A2141/datastream/PDF/view
[11] http://old.riteh.hr/nast/obrane/strucni_el/rok1_06072016/Filip_Kadum.pdf

- STL (engl. *Statement List*)
- SCL (engl. *Structured Control Language*)

3.2.1. Function Block Diagram (FBD)

FBD koristi se za PLC programe opisane u grafičkim blokovima (Slika 3.4). Zamišljen je kao grafički jezik za prikazivanje signala i protoka podataka kroz blokove, koji su višestruko upotrebljivi elementi softvera. Funkcijski blok je programska instrukcijska jedinica koja, kada se izvršava, daje jednu ili više izlaznih vrijednosti[17].

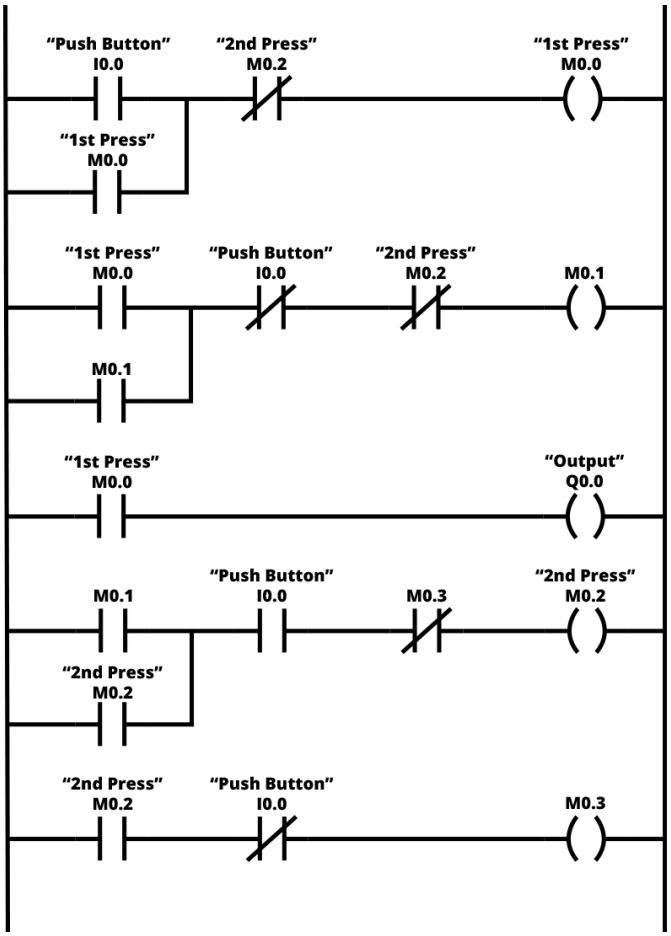


***Slika 3.4**:Primjer FBD-a*[12]

3.2.2. Ladder Logic (LAD)

Od različitih jezika koji se mogu koristiti za programiranje PLC-a, LAD je jedini koji je izravno modeliran prema elektromehaničkim relejnim sustavima. Koristi duge prečke postavljene između dvije vertikalne šipke koje predstavljaju napajanje sustava (Slika 3.5). Uz prečke su kontakti i svici, oblikovani po kontaktima i zavojnicama na mehaničkim relejima. Kontakti djeluju kao ulazi i često predstavljaju prekidače ili tipke. Zavojnice se ponašaju kao izlazi kao što su svjetlo ili motor[18].

[12] https://www.sciencedirect.com/topics/computer-science/function-block-diagram



***Slika 3.5**:Primjer LAD-a*[13]

3.2.3. Statement list (STL)

STL je tekstualni programski jezik koji se može koristiti za izradu sekcije koda logičkih blokova. Njegova sintaksa slična je jeziku asemblera i sastoji se od naredba koje slijede adrese na kojima djeluju upute (Slika 3.6). Od svih programskih jezika s kojima se mogu programirati Step7 kontroleri, STL je najbliži strojnom kodu. STL ima sve potrebne elemente za izradu kompletnog korisničkog programa. Sadrži opsežan raspon uputa. Dostupno je ukupno više od 130 različitih osnovnih uputa i širok raspon adresa[19].

[13] https://www.plcacademy.com/ladder-logic-examples/

```
Network 1: Title:
        A(
        A(
        CALL  "Sine_signal"                                  FC1              -- test
         Amplitude:="signal generator".Inputs.Amplitude1     DB100.DBD0
         Frequency:="signal generator".Inputs.frequency1     DB100.DBD24
         Val_Sinus:="signal generator".Outputs.signal1       DB100.DBD38
        A     BR
        )
        JNB   _001
        CALL  "Sine_signal"                                  FC1              -- test
         Amplitude:="signal generator".Inputs.amplitude2     DB100.DBD4
         Frequency:="signal generator".Inputs.Frequency2     DB100.DBD28
         Val_Sinus:="signal generator".Outputs.signal2       DB100.DBD42
_001:   A     BR
        )
        JNB   _002
        CALL  "Sine_signal"                                  FC1              -- test
         Amplitude:="signal generator".Inputs.amplitude3     DB100.DBD8
         Frequency:="signal generator".Inputs.frrquency3     DB100.DBD32
         Val_Sinus:="signal generator".Outputs.signal3       DB100.DBD46
_002:   NOP   0
```

***Slika 3.6**:Primjer STL-a*

3.2.4. Structured Control Language (SCL)

SCL je tekstualni programski jezik više razine koji se temelji na PASCAL-u. Uz elemente jezika visoke razine, SCL također uključuje elemente jezika tipične za PLC-ove kao što su ulazi, izlazi, tajmeri, bit memorija, blok-pozivi itd (Slika 3.7). Za optimalno korištenje i praktičnu primjenu SCL-a, postoji snažno razvojno okruženje koje odgovara specifičnim karakteristikama SCL-a i Step-a 7[20].

```
FUNCTION FC10 : VOID

CONST
    FIRST_NAME := 'FirstName';
    LAST_NAME := 'LastName';
END_CONST

VAR_INPUT
    no: INT;
END_VAR

VAR_OUTPUT
    character: STRING;
END_VAR

IF no = 1
THEN
    character := FIRST_NAME;
ELSE
    character := LAST_NAME;
END_IF;

END_FUNCTION
```

***Slika 3.7**:Primjer SCL-a*[14]

[14]https://support.industry.siemens.com/cs/document/52258437/how-do-you-define-the-constants-in-an-s7-scl-program-in-step-7-(tia-portal)-?dti=0&lc=en-WW

3.3. Codesys

CoDeSys (engl. *Controller Development System*) kao i Step7 služi za programiranje PLC-ova prema IEC 61131-3 standardu. CODESYS razvija i prodaje njemačka softverska tvrtka 3S-Smart Software Solutions. Codesys, za razliku od Step-a 7 omogućuje vizualizaciju programa. Codesys podržava sljedeće programske jezike[21]:

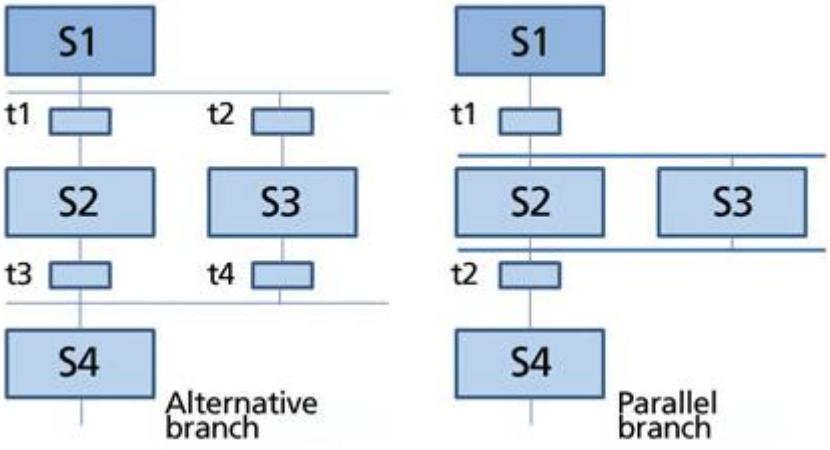
- IL (engl. *Instruction List*)
- ST (engl. *Structured Text*)
- LAD (engl. *Ladder Diagram*)
- FBD (engl. *Function Block Diagram*)
- SFC (engl. *Sequential Function Chart*)

Codesys također podržava CFC (engl. *Continuous Function Chart*), ali on nije definiran IEC 61131-3 standardom.

3.3.1. Programski jezici Codesys-a

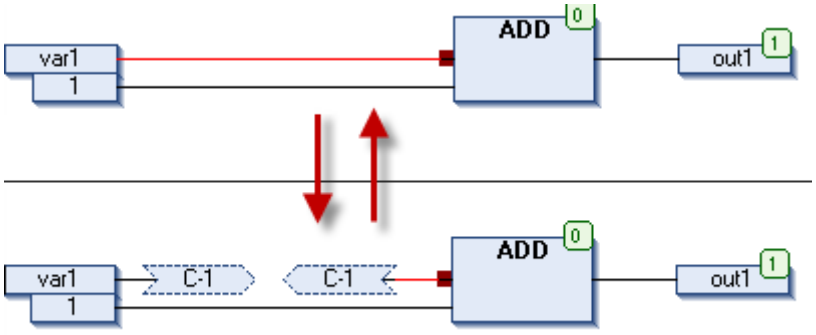
Instruction List je nalik asemblerskom jeziku, te ima sličnu sintaksu kao STL u Step-u 7. *Structured Text* je viši programski jezik koji je nalik SCL-u u Step-u 7.

SFC je grafički programski jezik (Slika 3.8). Ključni koncepti na kojima se temelje SFC-ovi su koraci i prijelazi. Korak je u osnovi neka funkcija unutar cjelokupnog sustava, poput pojedinačnog strojnog procesa. Prijelaz je upravo to, promjena iz jednog koraka u drugi korak ili stanje. Osim osnove, SFC programi mogu također uključivati standardne tehnike logičkog programiranja kao što su povratne petlje i grananje (paralelne ili alternativne grane.) SFC-i se također mogu dizajnirati pomoću dijagrama pomoćnih stanja[22].



***Slika 3.8**:Primjer SFC-a*[15]

CFC (Slika 3.9) dopušta programiranje bez privremenih varijabli i blokovi se mogu pozivati neovisno o drugima[21].



***Slika 3.9**:Primjer CFC-a*[16]

3.4.Tipovi podataka

Svaki tip podatka je definiran određenom dužinom u bitovima koju zauzima u memoriji PLC-a (Slika 3.10).

[15] https://www.motioncontroltips.com/sequential-function-charts-sfcs-plcs/
[16] https://repozitorij.etfos.hr/islandora/object/etfos%3A2141/datastream/PDF/view

Name	Type	Bits	Range
BOOL	boolean	1	0 to 1
SINT	short integer	8	-128 to 127
INT	integer	16	-32768 to 32767
DINT	double integer	32	-2.1e-9 to 2.1e9
LINT	long integer	64	-9.2e19 to 9.2e19
USINT	unsigned short integer	8	0 to 255
UINT	unsigned integer	16	0 to 65536
UDINT	unsigned double integer	32	0 to 4.3e9
ULINT	unsigned long integer	64	0 to 1.8e20
REAL	real numbers	32	
LREAL	long reals	64	
TIME	duration	not fixed	not fixed
DATE	date	not fixed	not fixed
TIME_OF_DAY, TOD	time	not fixed	not fixed
DATE_AND_TIME, DT	date and time	not fixed	not fixed
STRING	string	variable	variable
BYTE	8 bits	8	NA
WORD	16 bits	16	NA
DWORD	32 bits	32	NA
LWORD	64 bits	64	NA

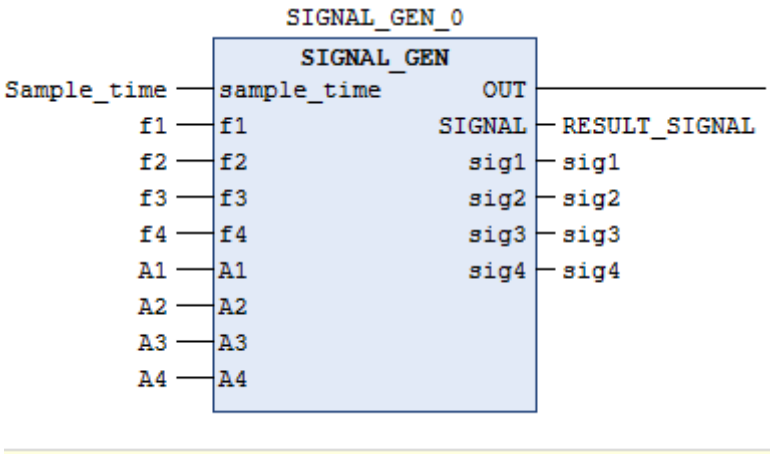
***Slika 3.10**:Tipovi podataka*[17]

[17] http://engineeronadisk.com/V3/engineeronadisk-160.html

4. PROGRAMSKO RJEŠENJE

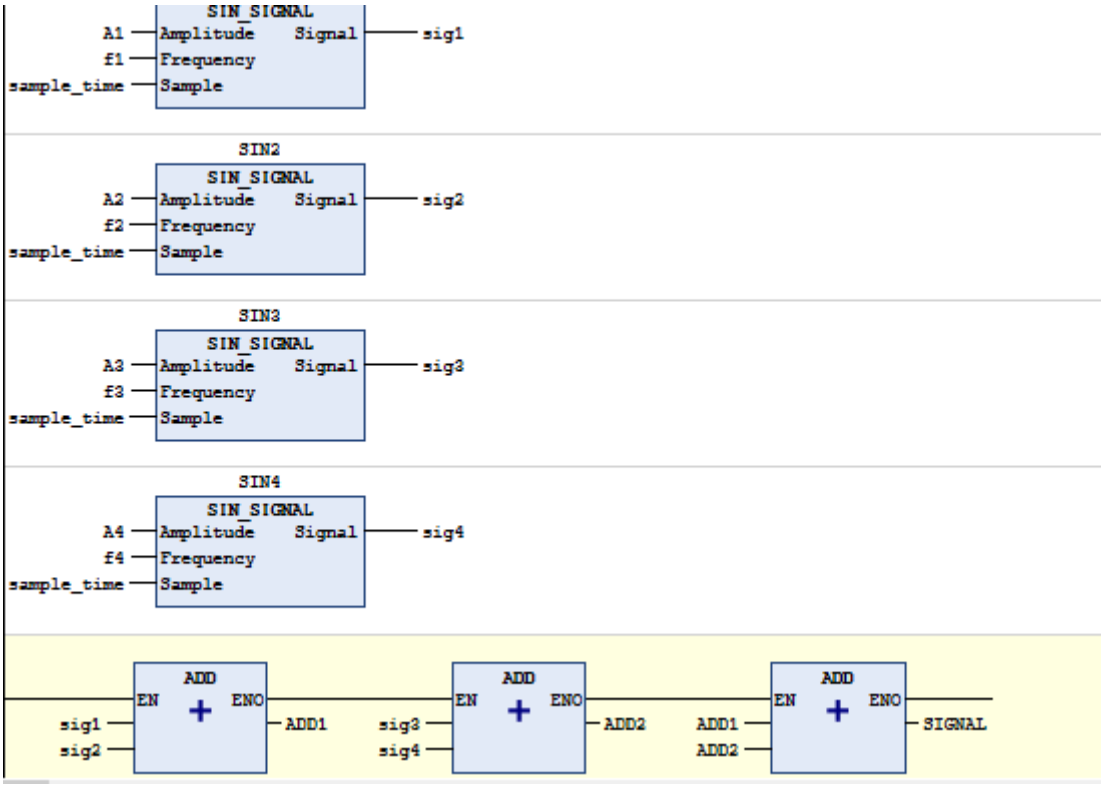
4.1. Programsko rješenje u Codesys-u

Programsko rješenje je pisano u LAD-u (engl. *Ladder Logic*) i u ST-u (engl. *Structured Text*). Prije nego što se može krenuti primjenjivati Fourierovu analizu, treba se generirati signal koji će se analizirati.



Slika 4.1:*Funkcijski blok za generiranje ulaznog signala*

Kao što se vidi na slici 4.1, generator signala prima na ulazu četiri frekvencije i četiri amplitude, te vrijeme proteklo od prijašnjeg pozivanja bloka (vrijeme ciklusa). Vrijeme ciklusa može se ručno podešavati, u našem primjeru je korišteno vrijeme ciklusa 50 milisekundi. Svaka kombinacija frekvencije, vrijeme ciklusa i amplitude kreira jednostavni sinusni signal, koji se potom zbrajaju i kreiraju *RESULT_SIGNAL* (Slika 4.2). *Sig1*, *sig2*, *sig3*, *sig4* predstavljaju vrijednosti generiranih sinusnih signala.



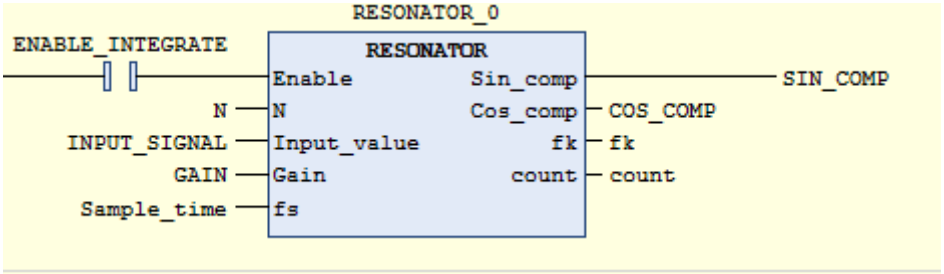
***Slika 4.2**:Prikaz funkcijskog bloka za sinusni signal i zbroj signala*

Funkcijski blok sinusnog signala, prima amplitudu, vrijeme ciklusa i frekvenciju te stvara jednostavni sinusni signal.

```
step := 6.28 * Sample * Frequency;
rad := rad + step;
IF rad > 6.28 OR rad < 0.0 THEN
    rad := 0.0;
END_IF

Signal := Amplitude * SIN(rad);
```

***Slika 4.3**:Kod za generiranje sinusnog signala*

Prema slici 4.3., *step* se dobiva tako da se pomnoži 2π, vrijeme ciklusa i frekvencija. Varijabla *rad* će se povećavati svaki ciklus za *step*, te ako je rad veći 2π ili manji od 0, postavlja ga se na nulu. Signal na kraju se dobiva tako da se pomnoži amplituda sa funkcijom *SIN*, koja prima radijane.



***Slika 4.4**:Fourierov rezonator*

Fourierova analiza vrši se u funkcijskom bloku *RESONATOR* (Slika 4.4). Na ulazu se nalazi pin *ENABLE_INTEGRATE*, to je bool tip podatka koji omogućava vršenje analize ako je postavljen na *TRUE*. Ako ga tijekom analize postavimo na *FALSE*, analiza se prekida i sve vrijednosti se postavljaju na nulu. Varijabla N predstavlja broj elemenata polja. Polje se sastoji od vrijednosti generiranog signala *RESULT_SIGNAL*. Ulaz *GAIN* služi da se smanjuju ili povećavaju vrijednosti naših izlaznih komponenti. *Sample_time* predstavlja vrijeme proteklo od prijašnjeg pozivanja bloka (vrijeme ciklusa). Na izlazu se nalaze *SIN_COMP* i *COS_COMP*. One predstavljaju sinus i kosinus komponente ulaznog signala (*RESULT_SIGNAL)*. Na izlazu se također nalazi *fk*, koja predstavlja vrijednost frekvencije [Hz] one sinusne komponente koja ima najveću amplitudu (frekvencija dominantne komponente ulaznog signala, te *count* prikazuje trenutni broj elemenata ulaznog signala sa kojima se vrši analiza (konačni broj elemenata je definiran varijablom N.

Za sam proces Fourieove analize korišten je DFT.

$$X_k = \sum_{n=0}^{N-1} x[n] e^{\frac{-jkn2\pi}{N}} \tag{4-1}$$

Te njegovim daljnjim raspisom dobije se:

$$X_k = X_{re}[k] + jX_{im}[k] \tag{4-2}$$

$$X_k = X_{re}[k] + jX_{im}[k] \tag{4-3}$$

$$X_{im}[k] = -\sum_{n=0}^{N-1} x[n] \sin\left(\frac{kn2\pi}{N}\right) \quad (4\text{-}4)$$

$X_{re}[k]$ i $X_{im}[k]$ predstavljat će nam sinus i kosinus komponentu (Slika 4.5).

```
FOR k:=0 TO N  BY 1 DO
    Xre[k]:=0;
    FOR i:=0 TO N-1  BY 1 DO
    Xre[k]:=(Xre[k]+(W[i]*COS(i*k*2*3.14/N)));
    END_FOR
    Xim[k]:=0;
    FOR i:=0 TO N-1 BY 1 DO
        Xim[k]:=(Xim[k]-(W[i]*SIN(i*k*2*3.14/N)));
    END_FOR
    Cos_comp:=Gain_re[k]:=Gain*Xre[k];
    Sin_comp:=Gain_Im[k]:=Gain*Xim[k];
    P[k]:=SQRT(Gain_re[k]*Gain_re[k]+Gain_Im[k]*Gain_Im[k]);
    freq[k]:=k*360/N;
END_FOR
END_IF
```

***Slika 4.5**:Kod za Fourierovu analizu*

Implementacija formule 4-3 i 4-4 dat će nam željene komponente. *W[i]* predstavlja vrijednosti *RESULT_SIGNAL-a* koja se mijenja svakom iteracijom petlje. *P[k]* predstavlja spektar signala.

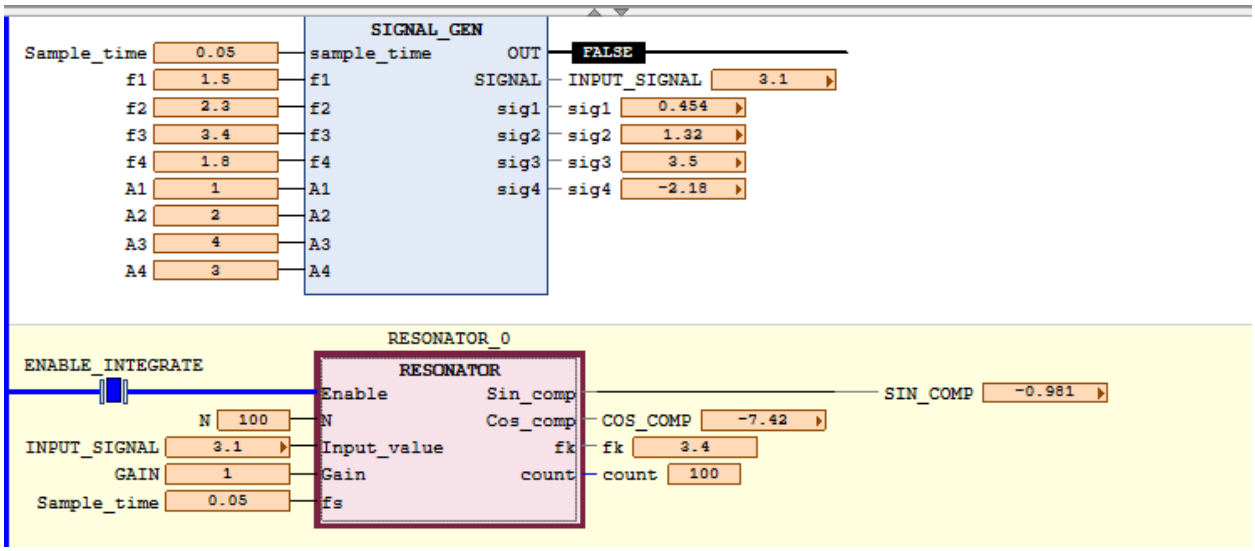
```
maximum:=0;
max_loc:=0;
FOR k:=1 TO n/2 BY 1 DO
    IF maximum <P[k] THEN
        maximum:=P[k];
        max_loc:=k;
    END_IF

END_FOR

fk:=max_loc/(N*fs);
```

***Slika 4.6**:Kod za pronalaženje fk*

Slika 4.6 predstavlja račun f_k. Prolazi se kroz vrijednost spektra signala i traži se lokacija najvećeg elementa. Ta vrijednost predstavlja amplitudu s najvećim utjecajem na *RESULT_SIGNAL*. Prolazi se samo kroz jedan dio spektra jer se u drugoj polovici vrijednosti ponavljaju zbog simetričnosti. Broj lokacije podijeli se s umnoškom broja elemenata vremena ciklusa. Pronalaskom ispravne vrijednosti f_k, dokazuje se da je Fourierova analiza uspješno izvršena.

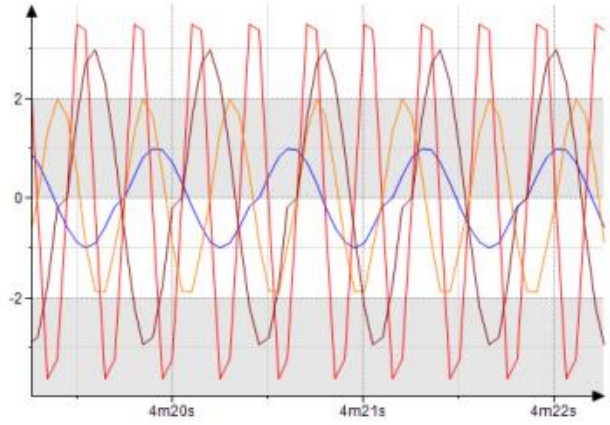
4.1.2. Vizualicaija

Codesys nudi mogućnost vizualizacije programa, tj. kreiranje HMI (engl. *Human Machine Interface*) sučelja. HMI je korisničko sučelje ili kontrolna ploča koja povezuje osobu sa strojem, sustavom ili uređajem.

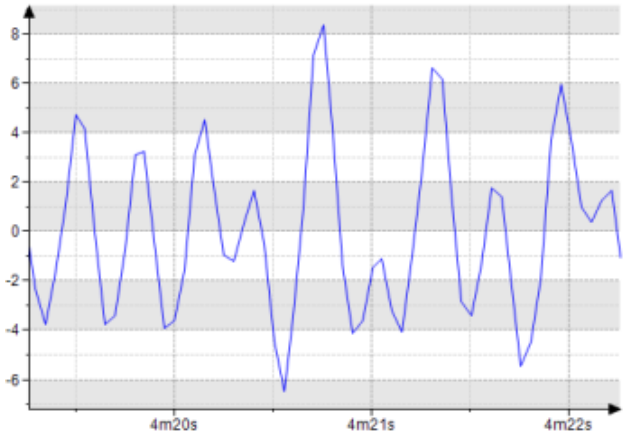


***Slika 4.7**:Funkcijski blokovi u simulaciji*

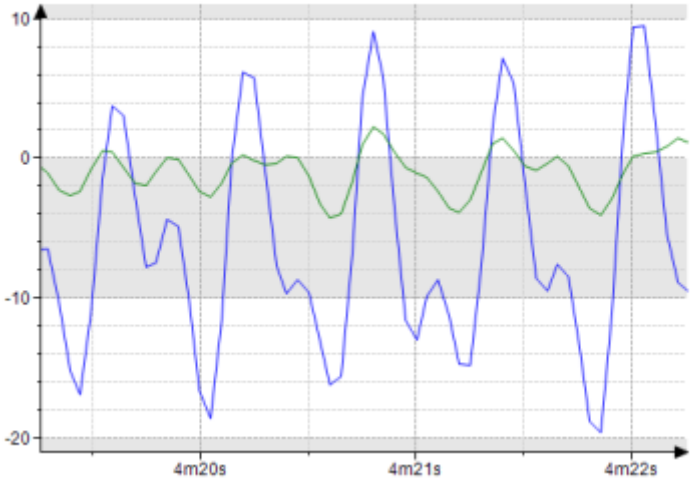
Kao što se vidi na slici 4.7, najveća amplituda je A3 i njena frekvencija iznosi 3.4 herca. Nakon što se provela analiza dobilo se $f_k = 3.4$ što odgovara frekvenciji najveće amplitude. *GAIN* je postavljen na 1. Nakon izvođenja prethodno objašnjenog programa dobiju se slijedeći grafički prikazi signala(Slika 4.8, Slika 4.9, Slika 4.10):



***Slika 4.8**:Generirani signali (sastavljen od 4 komponente)*



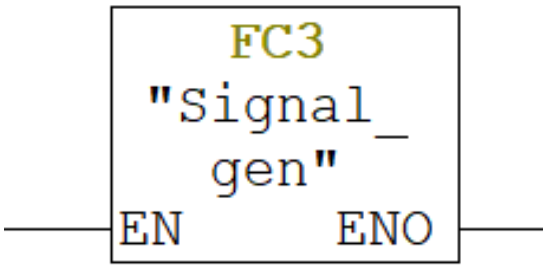
***Slika 4.9**:Zbroj signala sa slike 4.8 (rezultantni signal)*



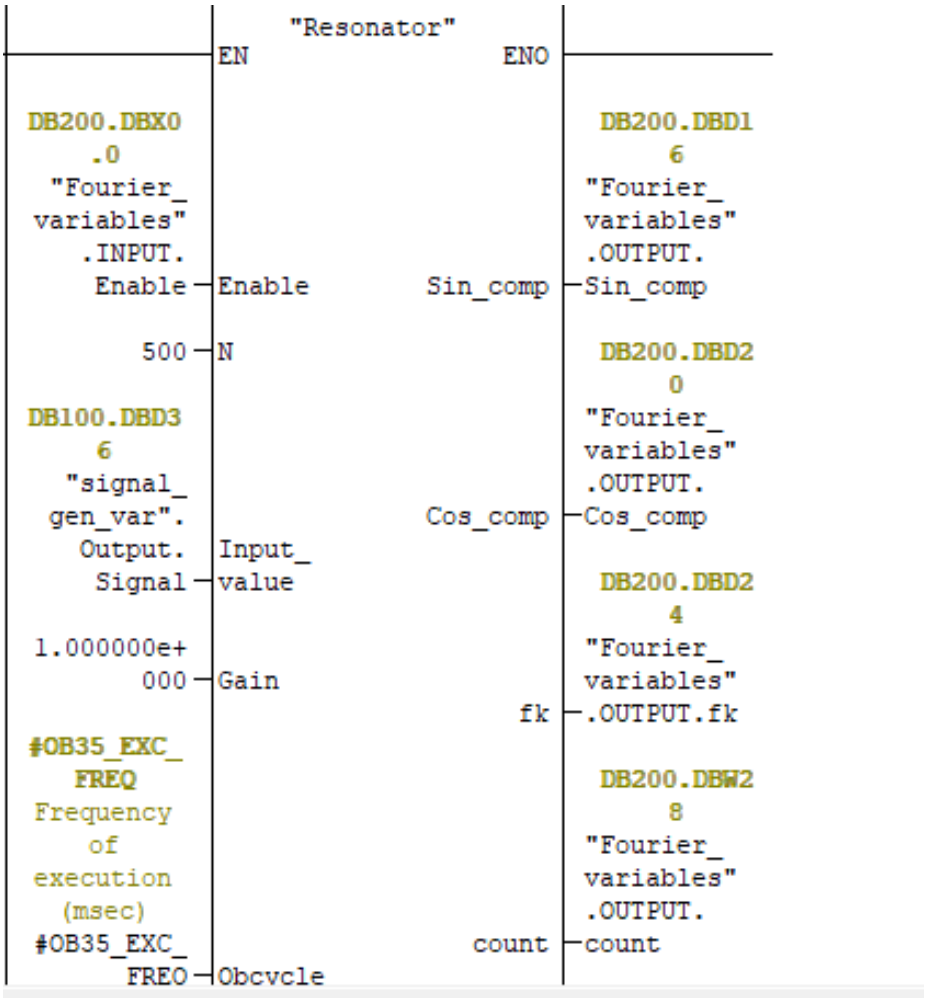
***Slika 4.10**:Kosinus (plavi signal) i Sinus (zeleni signal) komponenta*

4.2. Programsko rješenje u STEP7 alatu

Programsko rješenje je implementirano kao u Codesys-u. Korišten je LAD i SCL za rješenje. Ručno je namješteno izvršavanje ciklusa na 100 milisekundi. Signal se generira u OB1, a Fourierov rezonator postavljen je u koji se izvršava preko hardverskog „interupt-a“ koji je moguće definirati u postavkama samog kontrolera (u našem slučaju 100ms). Funcijski blokovi prikazani su na slici 4.11 i 4.12.



***Slika 4.11**:Funkcijski blok za generiranje signala*



***Slika 4.12**:Funkcijski blok za rezonator*

5. ZAKLJUČAK

Fourierova analiza je prisutna u gotovo svim područjima znanosti. Mogućnost rastavljanja signala na komponente koristi se u analizi signala kao što su slike, radio signal itd. Postoje razni oblici Fourierove transformacije, ali DFT se koristi najviše. FFT algoritmi implementiraju DFT na najefikasniji način, te tako možemo iskoristiti puni potencijal Fourierove analize. PLC uređaji upravljaju procesom i odrađuju svoje zadatke u realnom vremenu. Izrađen je program koji će vršiti Fourierovu analizu pomoću DFT-a. Za programsko rješenje simuliran je rad PLC-a u Codesys-u i STEP7 alatu. Svi procesi generiranja signala, zbrajanja signala i njihov rastav na komponente pokazan je putem HMI sučelja u Codesys-u. Pronalaženjem frekvencije signala s najvećom amplitudom, dokazuje da je analiza uspješno izvršena.

Programsko rješenje je pisano u Ladder-u, ST-u i SCL-u.

LITERATURA

[1] http://www.people.fas.harvard.edu/~djmorin/waves/Fourier.pdf

Datum pristupa linku 23.06.2019

[2] https://www.akustika.hr/_download/repository/03-OI-SpektarSignala.pdf

Datum pristupa linku 24.06.2019

[3] https://www.cv.nrao.edu/course/astr534/FourierTransforms.html

Datum pristupa linku 24.06.2019

[4] Tomislav Petković, Branko Jeren i ostali, ZBIRKA RIJEŠENIH ZADATAKA IZ SIGNALA I SUSTAVA, Zagreb, 2017

[5] Neven Elezović, MATEMATIKA 3: Fourierov red i integral Laplacova transformacija, Zagreb, 2007

[6] http://lavica.fesb.unist.hr/mat1/predavanja/node143.html

Datum pristupa linku 25.06.2019.

[7]https://loomen.carnet.hr/pluginfile.php/2163840/mod_resource/content/1/Predavanje%2014.pdf

Datum pristupa linku 24.06.2019

[8] http://e.math.hr/math_e_article/br19/matijevic

Datum pristupa linku 25.06.2019

[9] http://lab425.fesb.hr/TINF/teorijainf_12.htm

Datum pristupa linku 26.06.2019

[10] https://en.wikipedia.org/wiki/Discrete_Fourier_transform

Datum pristupa linku 29.06.2019.

[11] https://en.wikipedia.org/wiki/Fast_Fourier_transform

Datum pristupa linku 29.06.2019

[12]https://loomen.carnet.hr/pluginfile.php/281092/mod_resource/content/1/Stara_Predavanja/SiS15.pdf

[13] Tomislav Horvat, BRZA FOURIEROVA TRANSFORMACIJA NA PROCESORU TMS320VC5505, ZAVRŠNI RAD br. 2111, Sveučilište u Zagrebu, Fakultet Elektrotehnike i Računarstva

[14] https://en.wikipedia.org/wiki/Discrete-time_Fourier_transform
Datum pristupa linku 29.06.2019

[15] Marin Šepac, Programljivi logički kontroleri(PLC), Završni rad, Sveučilište u Rijeci, Filozofski fakultet u Rijeci, odsjek za politehniku

[16] http://www.plccenter.cn/Siemens_Step7/bas00014.htm
Datum pristupa linku 26.06.2019

[17] https://www.sciencedirect.com/topics/computer-science/function-block-diagram
Datum pristupa linku 26.06.2019

[18] https://www.allaboutcircuits.com/technical-articles/ladder-logic-programmable-logic-controller/ Datum pristupa linku 26.06.2019

[19] SIMATIC Structured Control Language(SCL) for S7-300/S7-400 Programming,Manual

[20] https://automationforum.in/t/plc-programming-introduction-to-statement-list/296
Datum pristupa linku 27.06.2019

[21] Emil Blažević, GENERIRANJE REFERENTNOG SIGNALA S OBLIKA, Sveučilište u Osijeku, Fakultet Elektrotehnike, Računarstva i Informacijskih tehnologija

[22] https://www.motioncontroltips.com/sequential-function-charts-sfcs-plcs/
Datum pristupa linku 27.06.2019

SAŽETAK

In this thesis, a function block for the Fourier analysis is described. For the purpose of testing the resonator block a function block for generating the input signal is created . The Fourier analysis is performed by the Discrete Fourier Transform (DFT). The generated signal represents a sum of four sinusoidal signals. The program allows changing the frequency and amplitude of different signal forms. An HMI interface which displays the signal components has been made in Codesys

Ključne riječi: Fourier, DFT, Codesys, Step7, HMI, PLC

ABSTRACT

Fourier resonator

In this thesis, a function block for Fourier analysis and a block for generating the input signal to be analyzed has been made. The function block for generating the signal was made for the purpose of testing the resonator block. For the process of the Fourier analysis, DFT has been used. The generated signal is composed of a sum of four sinusoidal signals. The frequency and amplitude can be varied for different signal forms. An HMI interface has been made in Codesys where we can see the signal components

Keywords: Fourier, DFT, Codesys, Step7, HMI, PLC

ŽIVOTOPIS

Marko Grubeša rođen je 18.9.1997. godine u Osijeku. U Piškorevcima završio je osnovnu školu Matije Gubeca. Potom upisuje jezičnu gimnaziju Antuna Gustava Matoša u Đakovu. Gimnaziju je prolazio sa odličnim uspjehom. Tečno govori njemački i engleski jezik, te iz njemačkog jezika posjeduje diplomu s potvrđenom C1 razinom znanja. 2016. završava gimnaziju te upisuje Fakultet elektrotehnike, računarstva i informacijskih tehnologija u Osijeku, smjer računarstvo. Od izvannastavnih aktivnosti član je studentske organizacije za razmjenu stručnih praksi IAESTE. Tijekom 2019. bio je na praksi u Danieli Systec d.o.o, gdje je i izrađivao završni rad.