



AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE
Wydział Elektrotechniki, Automatyki, Informatyki i Inżynierii Biomedycznej

Projekt dyplomowy

*Implementacja i weryfikacja modelu silnika prądu stałego
z wykorzystaniem graficznego języka programowania.*

*Implementation and verification of a DC motor model using a
graphical programming language.*

Autor:	<i>Przemysław Herian</i>
Kierunek studiów:	<i>Mikroelektronika w technice i medycynie</i>
Opiekun pracy:	<i>dr inż. Anna Kozioł</i>

Kraków, 2020

Spis treści

I. Wstęp.....	5
II. Cel i zakres pracy.....	7
III. Część teoretyczna.....	9
3.1 Silnik DC.....	9
3.1.1 Parametry i charakterystyki silnika DC.....	10
3.1.2 Zasada działania i budowa.....	11
3.1.3 Sterowanie.....	12
3.2 Model matematyczny.....	14
3.2.1 Model matematyczny silnika DC.....	15
3.3 Systemy wbudowane.....	17
3.4 Systemy czasu rzeczywistego.....	18
3.5 MyRIO.....	18
3.6 LabVIEW.....	19
3.7 FPGA.....	20
IV. Część praktyczna.....	21
4.1 Budowa stanowiska kontrolno-pomiarowego.....	21
4.2 Oprogramowanie stanowiska.....	25
4.3 Implementacja modelu matematycznego silnika.....	29
4.4 Przeprowadzenie testów.....	32
V. Podsumowanie.....	35
VI. Bibliografia.....	37

I. Wstęp

Już w prehistorii ludzie próbowali opisywać otaczającą ich rzeczywistość wykorzystując liczby. Starożytni uczeni wyznawali pogląd, że aby poznać prawdziwą naturę świata należy ją badać przy użyciu matematyki. Analizowali oni struktury działań elementów otoczenia i zapisywali swoje obserwacje, aby stworzyć ich matematyczny opis.

Dziś projektując wszelkie układy, które mają mieć odzwierciedlenie w rzeczywistości, inżynierowie polegają na specjalistycznych programach komputerowych. W zależności od dziedziny, każdy z tych programów posiada bazę podstawowych obiektów, których działanie jest ściśle określone i można je symulować. Za każdym takim obiektem stoi jego model matematyczny, czyli zbiór równań, który opisuje jego zachowanie w zależności od parametrów, danych wejściowych lub warunków otoczenia.

Celem tej pracy było sprawdzenie, jak dokładnie model matematyczny silnika prądu stałego odwzorowuje pracę rzeczywistego silnika, porównując osiągi obu tych obiektów.

II. Cel i zakres pracy

Celem niniejszej pracy była implementacja modelu silnika prądu stałego oraz porównanie modelu z osiąganymi prezentowanymi przez rzeczywisty silnik. Aby przeprowadzić charakteryzację rzeczywistego silnika, niezbędna była budowa stanowiska kontrolno-pomiarowego. Umożliwiało to pomiar jego parametrów z częstotliwością ponad 100kHz. Wysoką integrację zbudowanego systemu zapewniło wykorzystanie wspólnego środowiska programistycznego dla wszystkich warstw systemu (FPGA, RT, Host) – graficznego języka programowania LabVIEW, oraz wykorzystanie jednej, zintegrowanej platformy sprzętowej od National Instruments.

Pierwszym krokiem w realizacji pracy był dobór charakteryzowanego silnika. Aby dokonać jego charakteryzacji, konieczne było zbudowanie stanowiska umożliwiającego sterowanie zapodawanymi na jego wejście poziomami napięcia, a także ich pomiar. Podczas pracy silnika należało zmierzyć takie wartości jak prędkość obrotowa i natężenie prądu, oraz zapewnić wysoką dokładność ich odczytu. Kolejnym krokiem w realizacji pracy było zaczerpnięcie informacji o modelu matematycznym silnika DC i zaimplementowanie go w aplikacji. Aby jego działanie odwzorowywało rzeczywistość, należało przeprowadzić pomiary na dobranym silniku, wyznaczyć charakteryzujące go parametry i zastosować je w zaimplementowanym modelu. W ramach pracy przygotowano program, który umożliwia użytkownikowi przeprowadzanie testów silnika oraz jego modelu z takimi samymi sygnałami sterującymi, oraz obserwację rezultatów. Wynikiem pracy było porównanie osiągnięć rzeczywistego silnika zmierzonych przy pomocy stanowiska pomiarowego z teoretycznymi osiąganymi obliczonymi przez model matematyczny tego obiektu.

III. Część teoretyczna

3.1 Silnik DC

Silnik elektryczny to w najprostszym ujęciu urządzenie zamieniające energię elektryczną na mechaniczną. W codziennym życiu znajdujemy ich zastosowanie na każdym kroku, od najmniejszych w zabawkach, napędach CD, szczoteczkach elektrycznych poprzez większe w narzędziach lub sprzętach kuchennych aż do tych wystarczająco mocnych do zasilania tramwajów, schodów ruchomych lub zyskujących teraz na popularności samochodów elektrycznych.

Historia silnika elektrycznego sięga blisko 200 lat. Zapoczątkowało ją w 1821 roku odkrycie przez Michaela Faradaya zasady działania przewodnika w polu magnetycznym, a pierwszy praktyczny silnik prądu stałego powstał już w 1834 roku.

Silniki elektryczne są bardzo zróżnicowane, w ich podstawowym podziale wyróżnia się silniki prądu stałego, zmiennego oraz sterowane impulsowo. Każdy z tych rodzajów ma jeszcze swoje podkategorie a cały czas powstają jeszcze nowe konstrukcje. W niniejszej pracy skupiono się na silniku prądu stałego z magnesem trwałym.

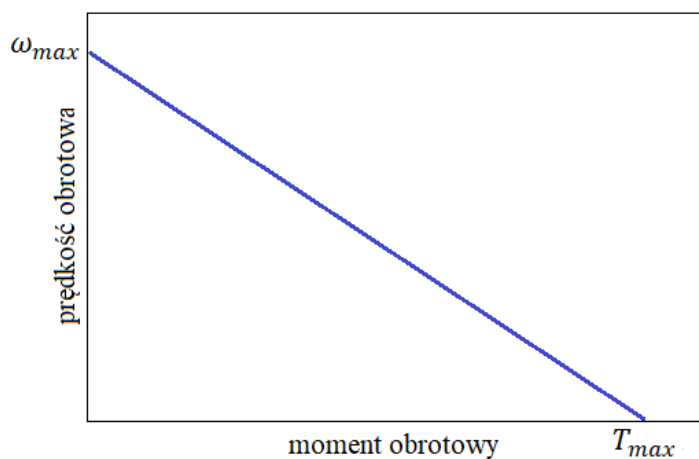
Silniki prądu stałego są wybierane z dwóch głównych powodów. Pierwszym powodem jest łatwość ich zasilania. Prąd stały jest najprostszy do otrzymania a przy zastosowaniach niestacjonarnych idealnie sprawują się wszędzie powszechne baterie lub akumulatory, z tego względu jest to jedyne rozwiązanie między innymi w motoryzacji. Drugim powodem, dla którego ludzie wybierają silniki prądu stałego do swoich aplikacji jest ich liniowa charakterystyka prędkości obrotowej od momentu obrotowego. W innych konstrukcjach zależność ta nie jest tak prosta do określenia, a nie znając dokładnych parametrów silnika wyznaczenie jej może przysporzyć dużych problemów.

3.1.1 Parametry i charakterystyki silnika DC

Podstawowe parametry określające działanie silnika, które są podawane przez producentów to:

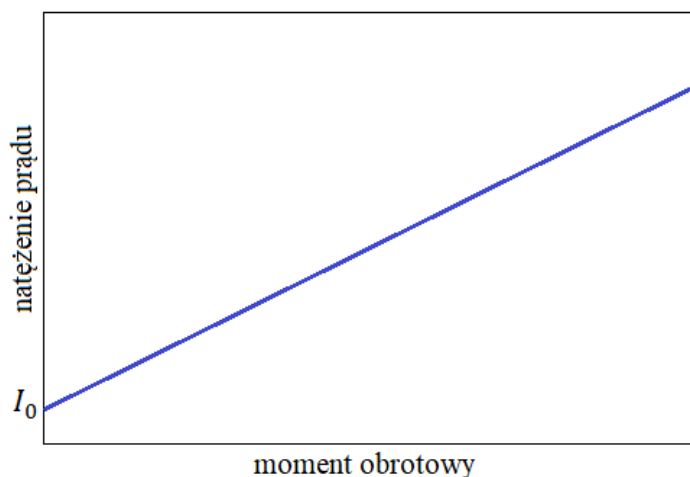
- znamionowe napięcie – wartość napięcia pracy, dla którego urządzenie zostało zaprojektowane i dla którego jest najbardziej wydajne, często maksymalne napięcie, pod wpływem którego element nie zostanie uszkodzony przy długotrwałym działaniu;
- znamionowa moc – moc jaka jest pobierana przy napięciu znamionowym podczas normalnej pracy silnika;
- znamionowy moment obrotowy – siła z jaką obraca się wirnik silnika przy napięciu znamionowym podczas normalnej pracy silnika;
- znamionowa prędkość obrotowa – prędkość obrotowa z jaką obraca się wirnik silnika przy znamionowym napięciu i ze znamionowym obciążeniem;
- znamionowy prąd – prąd pobierany przez silnik przy normalnych warunkach pracy;
- moment startowy – siła z jaką wirnik rozpoczyna obracanie, informuje z jakim maksymalnym obciążeniem jest w stanie wystartować silnik;
- prąd startowy – prąd pobierany przez silnik w trakcie jego rozpędzania, jest on dużo wyższy od prądu znamionowego.

Przy stałym napięciu zasilania wykres prędkości obrotowej od momentu obrotowego jest liniowy. Prędkość maleje wraz ze wzrostem obciążenia przyłożonego do wirnika



Rys. 1. Charakterystyka prędkości obrotowej od momentu obrotowego.

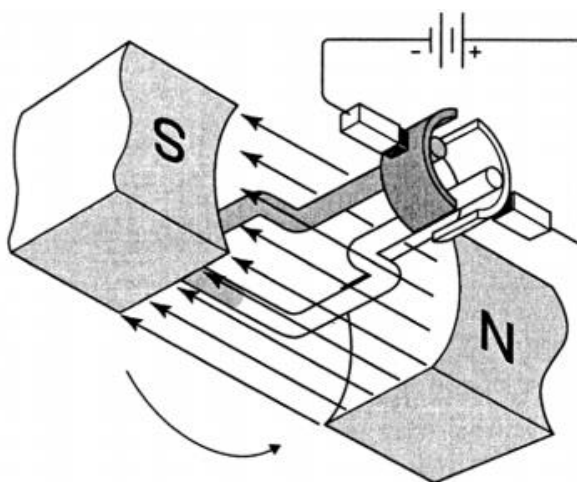
Natężenie pobieranego przez silnik prądu jest wprost proporcjonalne do jego momentu obrotowego.



Rys. 2. Charakterystyka natężenia prądu od momentu obrotowego.

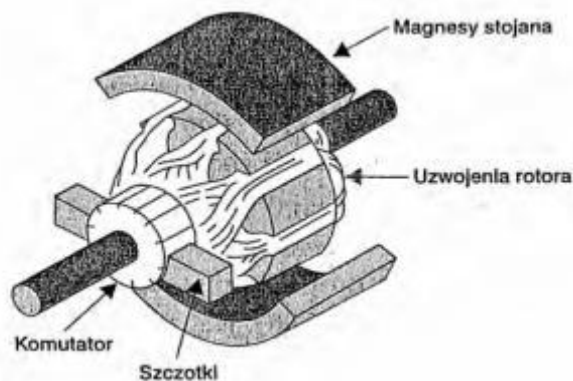
3.1.2 Zasada działania i budowa

Jeżeli w polu magnetycznym umieści się przewodnik, przez który przepływa prąd to będzie działała na niego elektrodynamiczna siła wypychająca z tego pola. Jeżeli w takim polu umieści się ramkę wewnątrz której płynie prąd, to na jej boki będą działały siły skierowane przeciwnie ponieważ kąt między kierunkiem prądu a kierunkiem linii pola magnetycznego zmienia się. Przy takiej konstrukcji, aby siły działające na ramkę nie zmieniały swojego kierunku co pół obrotu ramki, należy odpowiednio zmieniać kierunek przepływu prądu w ramce. W tym celu stosuje się komutator składający się z dwóch półpierścieni, do których prąd doprowadzany jest przez szczotki. Aby uzyskać równomierny obrót i większy wypadkowy moment obrotowy, zamiast jednej ramki stosuje się ich wiele i rozmieszcza symetrycznie, w ten sposób tworzy się twornik silnika.



Rys. 3. Schemat działania silnika prądu stałego. [1]

Powszechnie silniki prądu stałego konstruowane są ze stojana, na którego ścianach po przeciwnych stronach umieszcza się dwa magnesy zwrócone do siebie biegunami różnoimiennymi, ruchomego wirnika podłączonego do równomiernie rozmieszczonych uzwojeń twornika, wraz z komutatorem oraz szczotek zasilających.

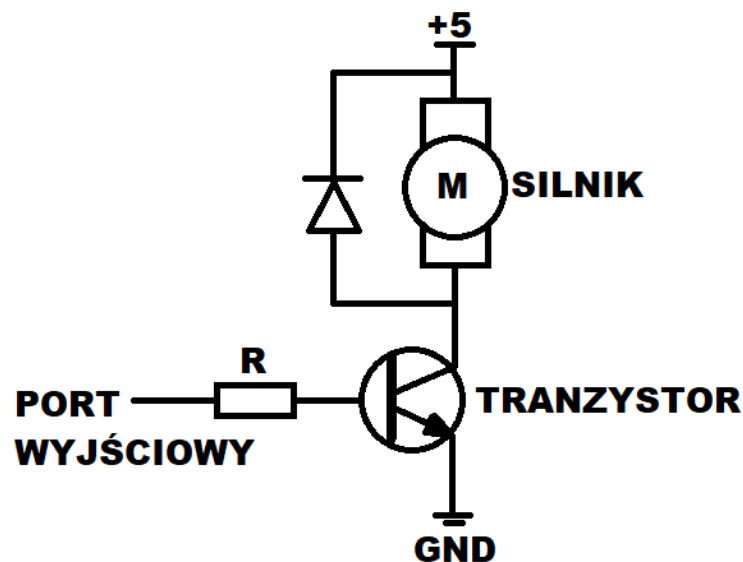


Rys. 4. Schemat budowy silnika prądu stałego. [2]

3.1.3 Sterowanie

KLUCZ TRANZYSTOROWY

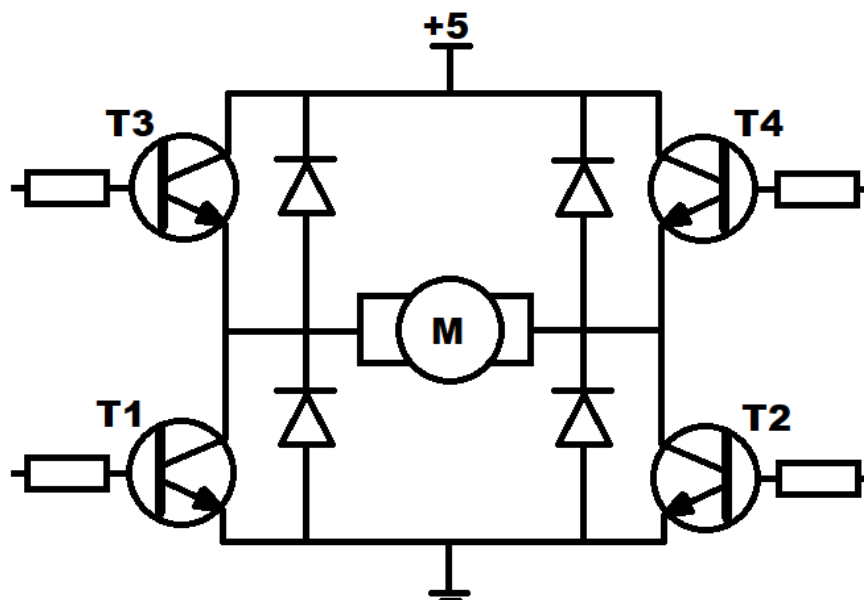
W przypadku sterowania silnika za pomocą mikrokontrolera może wystąpić problem z dostarczeniem do jego zacisków prądu o odpowiednio wysokim natężeniu. Wyjścia mikrokontrolera mają zwykle wydajność prądową na poziomie kilkudziesięciu miliamperów czyli dużo mniej niż jest to wymagane przez większość silników do pracy. Jeżeli próbowano by bezpośrednio kontrolować silnik to w najlepszym wypadku nie zareagowałby on na nadawane sygnały, taka próba mogłaby również uszkodzić mikrokontroler, który nie jest zabezpieczony na taką ewentualność. Z tego powodu należy zastosować pośredni układ, który umożliwi wysterowanie potrzebnego prądu. Najprostszym przykładem jest pojedynczy klucz tranzystora bipolarnego. Port wyjściowy mikrokontrolera podłącza się przez rezystor do bazy tranzystora, rezystor stosuje się aby ograniczyć prąd więc należy dobrać jego wartość tak, aby maksymalny przepływający przez niego prąd nie był większy niż maksymalny prąd, który może dostarczyć wyjście mikrokontrolera. Jeden z zacisków silnika przyłącza się do stałego źródła napięcia, a drugi do kolektora tranzystora. Emiter podłącza się do masy. Układ taki działa jako przełącznik, pojawienie się na bazie tranzystora stanu wysokiego, spowoduje przepływ prądu przez silnik i jego załączenie. Koniecznym jest dodatkowo dodanie diody między zaciski silnika. Po wyłączeniu tranzystora prąd nie przestaje płynąć od razu lecz stopniowo zanika. Przez indukcyjność silnika zostaje generowane napięcie samoindukcji, które proporcjonalne jest do tempa zmniejszania się prądu. Energia zgromadzona w indukcyjności zostaje przez diodę bezpiecznie rozładowana nie uszkadzając tranzystora.



Rys. 5. Schemat sterowania kluczem tranzystorowym.

MOSTEK H

W sterowaniu silnikiem wymagana jest często kontrola kierunku jego obrotów. Możliwości takiej nie ma przy sterowaniu za pomocą pojedynczego klucza tranzystorowego, ponieważ zawsze ten sam zacisk jest podłączony do zasilającego napięcia. Z pomocą przychodzi tu układ mostka H. Nazwa tego układu wzięła się od wyglądu jego schematu. Składa się on z 4 tranzystorów i 4 diod, których zastosowanie jest analogiczne jak w przypadku pojedynczego klucza tranzystorowego.



Rys. 6. Schemat układu mostka H.

Mostek H ma 4 tryby pracy:

- gdy przewodzące są tranzystory T3 i T2 a zatkane T1 i T4 prąd płynie od źródła napięcia przez T3, silnik, T2 do masy - silnik obracany jest w wybraną stronę;
- gdy przewodzące są tranzystory T1 i T4 a zatkane T2 i T3 prąd płynie od źródła napięcia przez T4, silnik i T1 do masy - silnik obracany jest w stronę przeciwną;
- gdy przewodzące są tranzystory T1 i T2 a zatkane T3 i T4 lub gdy przewodzące są tranzystory T3 i T4 a zatkane T1 i T2 - różnica napięcia na zaciskach silnika jest zerowa i uzyskujemy efekt hamowania silnika;
- gdy zatkane są wszystkie tranzystory silnik jest w trybie swobodnego obracania i powoli wytraca prędkość obrotową.

PWM

Są dwa główne sposoby na regulację napięcia sterującego silnikiem. Pierwszy to podłączenie go do stałego źródła napięcia przez potencjometr. Poprzez zmiany oporności potencjometru możemy kontrolować jaka część napięcia zasilania odłoży się na jego zaciskach, a co za tym idzie jakie napięcie zostanie dostarczone do silnika. Wadą tego rozwiązania są duże straty mocy, jeżeli chcielibyśmy uzyskać niską prędkość obrotową to energia stracona na oporniku byłaby większa od tej pobieranej przez silnik. Zdecydowanie lepszym sposobem jest sterowanie sygnałem prostokątnym o stałej amplitudzie napięcia i modulacja szerokości impulsu (PWM). Sygnał taki możemy uzyskać używając przełącznika tranzystorowego jaki został opisany wcześniej. Zwykle przy PWM stosuje się różne filtry, aby uśrednić i wygładzić przebieg napięcia. Dzięki specyfikacji silnika DC nie jest potrzebne ich stosowanie. W niskich częstotliwościach sygnału prostokątnego odpowiada za to jego bezwładność mechaniczna, a w wysokich indukcyjność. Jeżeli chodzi o dobór odpowiedniej częstotliwości sterowania to przyjmuje się, że przy maksymalnej pożądanej prędkości obrotowej na każdy obrót silnika powinno przypadać od 2 do 5 impulsów, a więc dla większości niedużych silników ta wartość leży w przedziale od stu do kilku tysięcy herców. Zbyt mała częstotliwość sprawi, że przebieg napięcia nie będzie wystarczająco wygładzony i w pracy silnika mogą pojawić się szarpania. Zbyt duża częstotliwość spowoduje natomiast większe straty mocy przy przełączaniu tranzystora.

3.2 Model matematyczny

Model matematyczny to reprezentacja badanego obiektu jako układu równań. Dzięki przeprowadzeniu procesu poznawczego można scharakteryzować zachowanie obiektu w zależności od czynników na niego wpływających. Model pozwala odwzorować wyidealizowane działanie rzeczywistego układu. Model znajduje zastosowanie w każdej dziedzinie, w której zachodzi powtarzalność, a więc zarówno w fizyce przy analizowaniu np. działania układów elektrycznych lub mechanicznych, jak również w medycynie przy opisie działania leku na organizm pacjenta lub w socjologii do określenia zachowania grup społecznych.

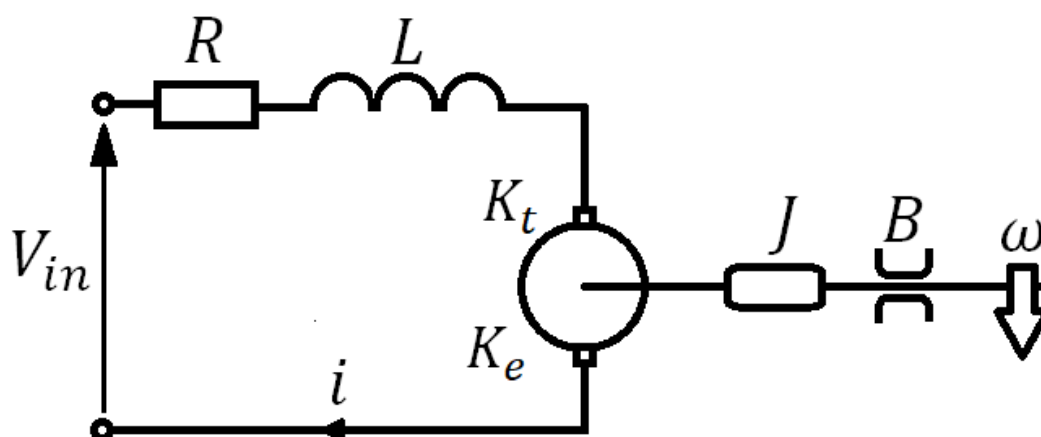
Modelowanie jest niezbędne w inżynierii do projektowania każdego układu od prostych układów elektrycznych i mechanicznych po np. całe budynki lub samoloty. Pozwala ono na teoretyczne wypróbowanie działania układu i dostosowania go do własnych potrzeb.

Model powstaje w 3 podstawowych krokach:

- Pierwszym krokiem jest obserwacja. Zbieranie informacji o układzie i jego zachowaniu. Obserwacja może być bezpośrednia – przez nasze zmysły, lub pośrednia w wypadku gdy przeprowadzane są pomiary aby stwierdzić czy zaszło jakieś zdarzenie.
- Następnie można przystąpić do analizowania zebranych obserwacji. Celem powstania modelu może być opis zachowania, wyjaśnienie dlaczego określone zachowanie zachodzi, lub jego przewidywanie.
- Trzecim krokiem jest użycie powstałego modelu do określenia przyszłego zachowania układu. Przewidywania są porównywane z dalszymi obserwacjami by zweryfikować poprawność modelu. Proces jest powtarzany do momentu uzyskania satysfakcjonującego odzwierciedlenia systemu.

3.2.1 Model matematyczny silnika DC

Zdecydowana większość siły napędowej w robotyce pochodzi z silników prądu stałego. Aby zasymulować działanie takiego silnika potrzebne jest najpierw określenie jego modelu. Pomoże to dobrać napięcie i prąd do wymaganej prędkości obrotowej.



Rys. 7. Schemat zastępczy silnika prądu stałego.

Najważniejsze wielkości charakteryzujące silnik prądu stałego to:

- V_{in} – napięcie zasilania;
- i – prąd płynący przez uzwojenia;
- R – rezystancja uzwojeń;
- L – indukcyjność uzwojeń;
- K_t – stała momentu obrotowego;
- K_e – stała elektromotoryczna;
- B – stała współczynnika tarcia;
- J – moment bezwładności;
- T – moment obrotowy;
- ω – prędkość obrotowa;
- V_e – indukowane napięcie.

Z napięciowego prawa Kirchhoffa wynika:

$$V_{in} = V_R + V_L + V_e$$

$$V_R = I * R$$

$$V_L = L * \frac{dI}{dt}$$

Kiedy wirnik silnika obraca się wewnątrz jego uzwojeń indukuje się siła elektromotoryczna indukcji proporcjonalna do prędkości obrotowej:

$$V_e = K_e * \omega$$

Podstawiając pod pierwsze równanie napięcie zasilania równe jest:

$$V_{in} = I * R + L * \frac{dI}{dt} + K_e * \omega$$

Moment obrotowy silnika jest równy sumie sił przeciwstawiających się jego ruchowi i jest proporcjonalny do płynącego w nim prądu:

$$T = J * \frac{d\omega}{dt} + B * \omega$$

$$T = K_t * i$$

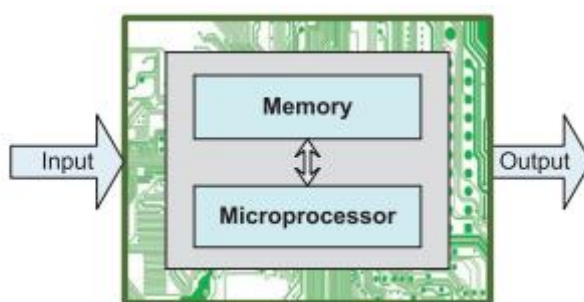
Otrzymujemy układ 2 równań, które opisują działanie silnika prądu stałego:

$$\begin{cases} V_{in} = I * R + L * \frac{dI}{dt} + K_e * \omega \\ K_t * i = J * \frac{d\omega}{dt} + B * \omega \end{cases}$$

3.3 Systemy wbudowane

Systemem wbudowanym określa się system elektroniczny, który zaprojektowany jest do wykonywania ściśle zdefiniowanych zadań i który często jest częścią większego układu. Różni się on od wielofunkcyjnych komputerów głównie pod dwoma względami:

- po pierwsze system wbudowany ma wypełniać dokładnie określoną funkcjonalność, podczas gdy urządzenia takie jak smartfon lub komputer osobisty mogą mieć bardzo szerokie zastosowania np. korzystanie z przeglądarek internetowych lub uruchamianie gier;
- po drugie zwykle w procesie wytwarzania systemu wbudowanego powstaje od razu także dedykowane oprogramowanie specjalnie dla tego urządzenia.



Rys. 8. Architektura systemu wbudowanego. [3]

System wbudowany, jest oparty na mikroprocesorze oraz bloku pamięci i jest połączeniem sprzętu komputerowego oraz oprogramowania, a czasami również mechanicznych elementów. Kiedy zostaje uruchomiony mikroprocesor czyta z pamięci zawarte w oprogramowaniu instrukcje, przetwarza zgodnie z nimi wejściowe informacje takie jak np. sygnały oraz dane z czujników i zwraca wynik.

Z reguły przy projektowaniu systemu wbudowanego większą uwagę przykładą się do napisania oprogramowania, na które składa się aplikacja, odpowiednie sterowniki i czasami system operacyjny. W wielu przypadkach możliwe jest stworzenie układu scalonego o równoważnej funkcjonalności, który nie zawiera oprogramowania ani mikroprocesora. Rzadko jest to jednak opłacalne z uwagi na to, że często występuje konieczność przystosowania działania systemu do określonych warunków. W takich przypadkach należy zaprojektować i wytworzyć nowy układ scalony, gdzie w przypadku systemu wbudowanego wystarcza aktualizacja oprogramowania.

3.4 Systemy czasu rzeczywistego

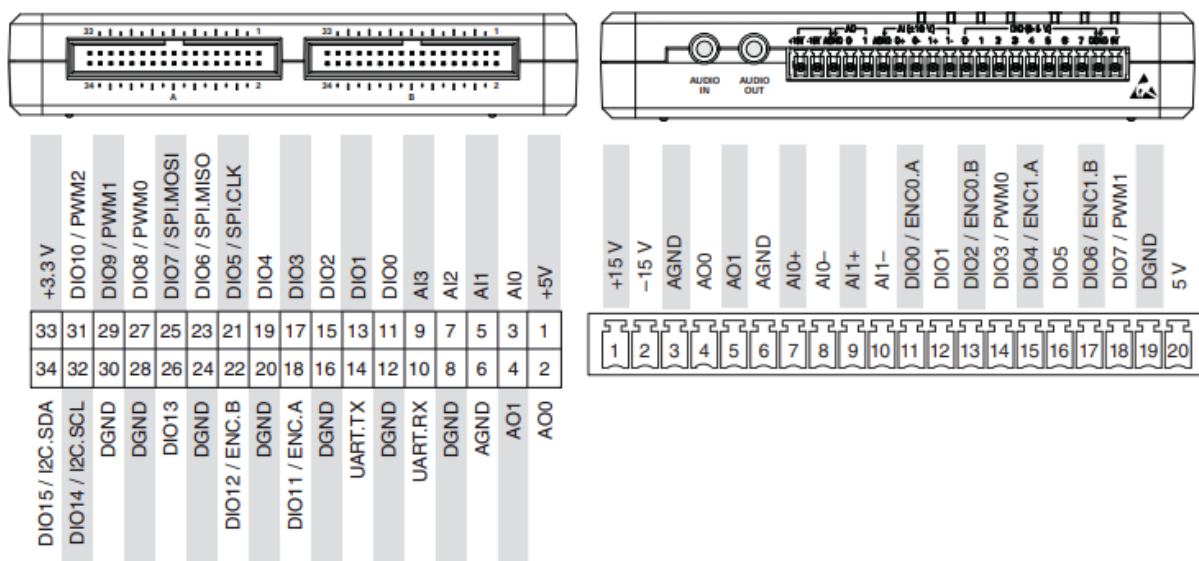
Od niektórych systemów wymagana jest reakcja w ściśle określonym czasie, systemy takie nazywane są systemami czasu rzeczywistego. Z uwagi na to jak krytyczne jest wykonanie zadania w wymaganym czasie, ograniczenia czasowe dzielimy na ostre i łagodne:

- ograniczenie czasowe jest ostre, jeżeli jego pogwałcenie jest krytyczne i spóźniona odpowiedź jest bezużyteczna lub nieakceptowalna;
- ograniczenie czasowe jest łagodne, gdy konsekwencje opóźnionej odpowiedzi systemu nie są tak poważne i można je zaakceptować jeżeli np. pojawiają się okazjonalnie i z małym prawdopodobieństwem.

Rzeczywiste systemy mogą mieć jednocześnie ostre i łagodne ograniczenia czasowe. Jeżeli wszystkie ograniczenia zadań w systemie są łagodne, to nazywa się go łagodnym systemem czasu rzeczywistego. Ostro system czasu rzeczywistego występuje wtedy, gdy jego kluczowe zadania mają ostre ograniczenia czasowe.

3.5 myRIO

National Instruments myRIO-1900 jest przenośnym rekonfigurowalnym urządzeniem wejścia-wyjścia. Zawiera ono zestaw portów analogowych i cyfrowych, moduł FPGA, wyjście zasilające jak i wyjście audio, akcelerometr oraz zestaw diod LED. Działanie wszystkich tych elementów można dostosowywać do wielu potrzeb. Można je połączyć z komputerem za pomocą USB. Dedykowanym językiem programowania przy pomocy, którego można konfigurować urządzenie jest LabVIEW ale można też wykorzystać C i C++.



Rys. 9. Układ pinów myRIO. [5]

3.6 LabVIEW

LabVIEW jest graficznym językiem programowania. Oznacza to, że w odróżnieniu od większości popularnych języków, linijki tekstu, którymi definiuje się działanie aplikacji są zastąpione ikonami umieszczanymi na diagramie. Architektura ta, eliminuje problemy składniowe występujące w tych tekstowych językach programowania.

Programy napisane za pomocą LabVIEW nazywa się Virtual Instruments (VI) ponieważ ich funkcjonalności często imitują rzeczywiste przyrządy. VI składa się z trzech podstawowych elementów: panelu frontowego, diagramu blokowego oraz ikony.

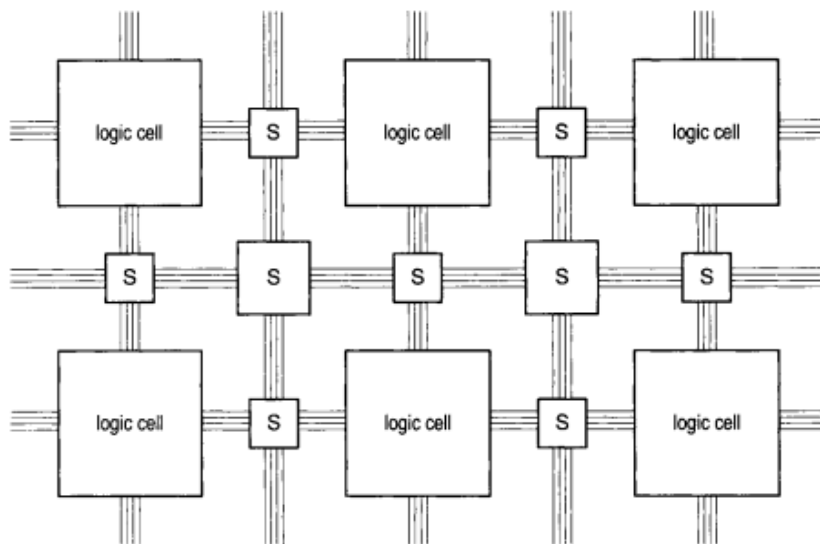
Panel frontowy jest interaktywnym interfejsem użytkownika. Został on tak nazwany ponieważ często odwzorowuje wygląd i działanie przyrządów pomiarowych np. oscyloskopu. Umieszczane są na nim kontrolki np. przyciski, suwaki i pokrętła, którymi determinować można działanie aplikacji, jak i indykatory np. wykresy, które umożliwiają obserwację wyników programu.

Diagram blokowy przedstawia kod źródłowy programu. Jest on stworzony przez podrzędne funkcje. Można podzielić je na pierwotne, które mają tylko ikonę i nie można ich edytować oraz na subVi'e, które posiadają wszystkie wcześniej wymienione elementy i mogą być tworzone i edytowane przez użytkowników. Funkcje łączy się przewodami i w ten sposób zapewnia przepływ danych. Ich wykonanie następuje w momencie kiedy otrzymują wystarczającą ilość danych wejściowych, a wynik działania wysyłają dalej.

Aby wykorzystać VI'a jako podprogram musi on także posiadać ikonę z obszarem połączeń. Ikona zawiera graficzną reprezentację funkcji, a obszar połączeń informuje do którego terminala powinno się podłączyć jakie dane dla prawidłowego działania programu. Terminale są odwzorowaniem kontrolki i indykatorów na panelu frontowym.

3.7 FPGA

Bezpośrednio programowalna macierz bramek (FPGA) to urządzenie logiczne zaprojektowane do konfiguracji przez użytkownika. Zawiera ono dwuwymiarową macierz komórek logicznych i programowalnych przełączników. Komórki logiczne mogą być konfigurowane tak, aby wykonywać prostą funkcjonalność, a przełączniki dostosowane aby zapewnić połączenia między nimi. Aby przygotować układ do konkretnego zadania należy określić funkcje wszystkich komórek logicznych i ustanowić połączenia między nimi. Po zaprojektowaniu połączeń wystarczy wgrać go na FPGA, aby otrzymać układ o pożądanym działaniu. Zaletą stosowania FPGA jest jego wydajność i szybkość. W przeciwieństwie do procesora możliwe jest równoległe przetwarzanie danych. Z tego powodu układy FPGA często znajdują zastosowanie w systemach czasu rzeczywistego.



Rys. 10. Struktura urządzenia FPGA. [4]

IV. Część praktyczna

Część praktyczna pracy składa się z czterech głównych zadań:

- Budowy stanowiska kontrolno-pomiarowego, doboru silnika oraz odpowiednich modułów, które umożliwią sterowanie silnikiem oraz przeprowadzenie wymaganych pomiarów;
- Odpowiedniego zaprogramowania obsługi stanowiska, prawidłowej interpretacji wysyłanych oraz odczytywanych pomiarów, zadbania o bezstratność i dokładność przesyłu danych oraz stworzenia interfejsu umożliwiającego użytkownikowi na łatwe przeprowadzenie testów;
- Interpretacji modelu matematycznego silnika i jego implementacji w wybranym środowisku, następnie przeprowadzeniu pomiarów oraz obliczeń na dobranym silniku do charakterystyki jego parametrów;
- Przeprowadzenia testów i porównania wyników rzeczywistego silnika odczytanych z pomocą zrealizowanego stanowiska oraz obliczonych przez zaimplementowany model matematyczny.

4.1 Budowa stanowiska kontrolno-pomiarowego.

Stanowisko kontrolno-pomiarowe jest zbudowane w oparciu o NI myRIO. Silnik jaki został wybrany to RF-300FA-12350. Wybór ten został podjęty ze względu na niskie napięcie zasilania, niezbyt wysoką prędkość obrotową, która nie sprawia problemu przy jej próbie odczytu oraz nakładkę na wale silnika, która ułatwia pomiary prędkości obrotowej oraz momentu obrotowego.

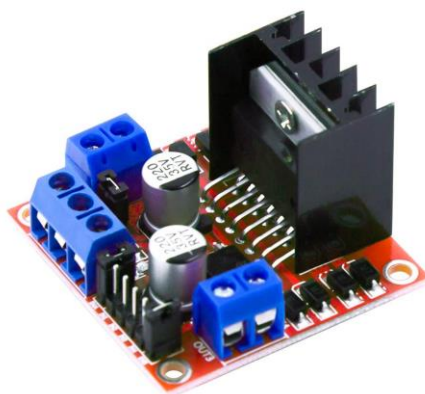


Rys. 11. Silnik RF-300FA. [7]

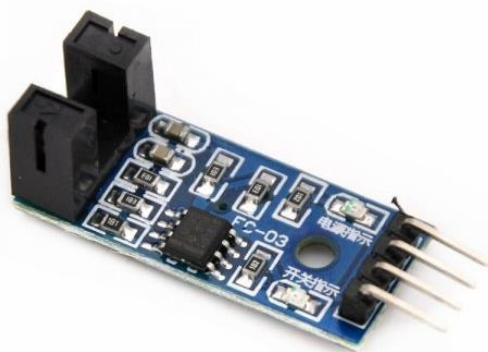
Tab. 1. Specyfikacja silnika RF-300FA. [6]

MODEL		VOLTAGE		NO LOAD		AT MAXIMUM EFFICIENCY					STALL		
		OPERATING RANGE	NOMINAL	SPEED	CURRENT	SPEED	CURRENT	TORQUE		OUTPUT	TORQUE		CURRENT
			V	r/min	A	r/min	A	mN·m	g·cm	W	mN·m	g·cm	A
RF-300FA	12350	1.5 - 6.0	3	3500	0.022	2830	0.093	0.48	4.9	0.14	2.51	26	0.39

Z powodu tego, że maksymalne natężenie prądu jakie jest możliwe do wyprowadzenia z portów sterujących myRIO to 20mA, a według tabeli parametrów podanych przez producenta silnik przy nominalnym napięciu i bez obciążenia wymaga 22mA silnik został podłączony do układu sterującego L298N z mostkiem H. Cześć logiczna układu została zasilona z wyjścia 5V myRIO a silnik z zewnętrznej baterii. Przed silnikiem dodatkowo wpięto szeregowo rezystor 9.1 Ω aby umożliwić wyznaczenie prądu przez pomiar napięcia.

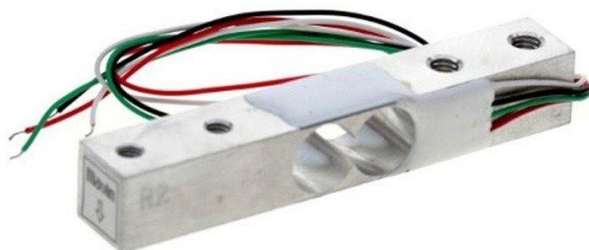
**Rys. 12. Układ L298N. [8]**

Do pomiaru prędkości obrotowej wykorzystano szczelinowy czujnik podczerwieni z układem komparatora LM393. Jest on zasilany stałym napięciem 5V z myRIO a wyjście cyfrowe jest bezpośrednio podłączone do portu wejściowego myRIO. Zasada jego działania to wystawianie na cyfrowe wyjście jedynki lub zera w zależności od tego czy sygnał z nadajnika dociera do odbiornika, czy jakiś obiekt znajduje się między nimi i blokuje przesył. Nakładka na wale silnika posiada otwory i z jej pomocą można przeprowadzić taki pomiar. Podczas działania silnika czujnik wykrywa kiedy między jego ramionami znajduje się otwór nakładki. Przez pomiar czasu między kolejnymi wykrytymi otworami można wyznaczyć ostatni okres obrotu, a co za tym idzie prędkość obrotu silnika w danym momencie.

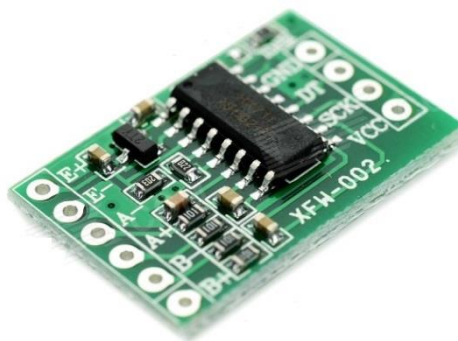


Rys. 13. Czujnik przerwań z układem komparatora LM393. [9]

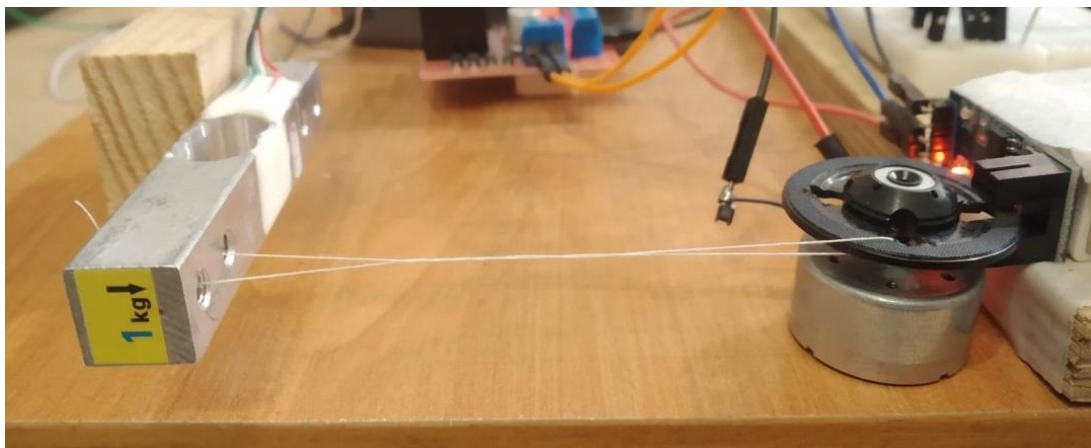
Aby przeprowadzić pomiary momentu obrotowego silnika zamontowano belkę tensometryczną NA27 działającą w zakresie do 10N i podłączono ją do myRIO za pomocą dedykowanego dla czujników tensometrycznych modułu układu HX711 z 24 bitowym przetwornikiem analogowo cyfrowym.



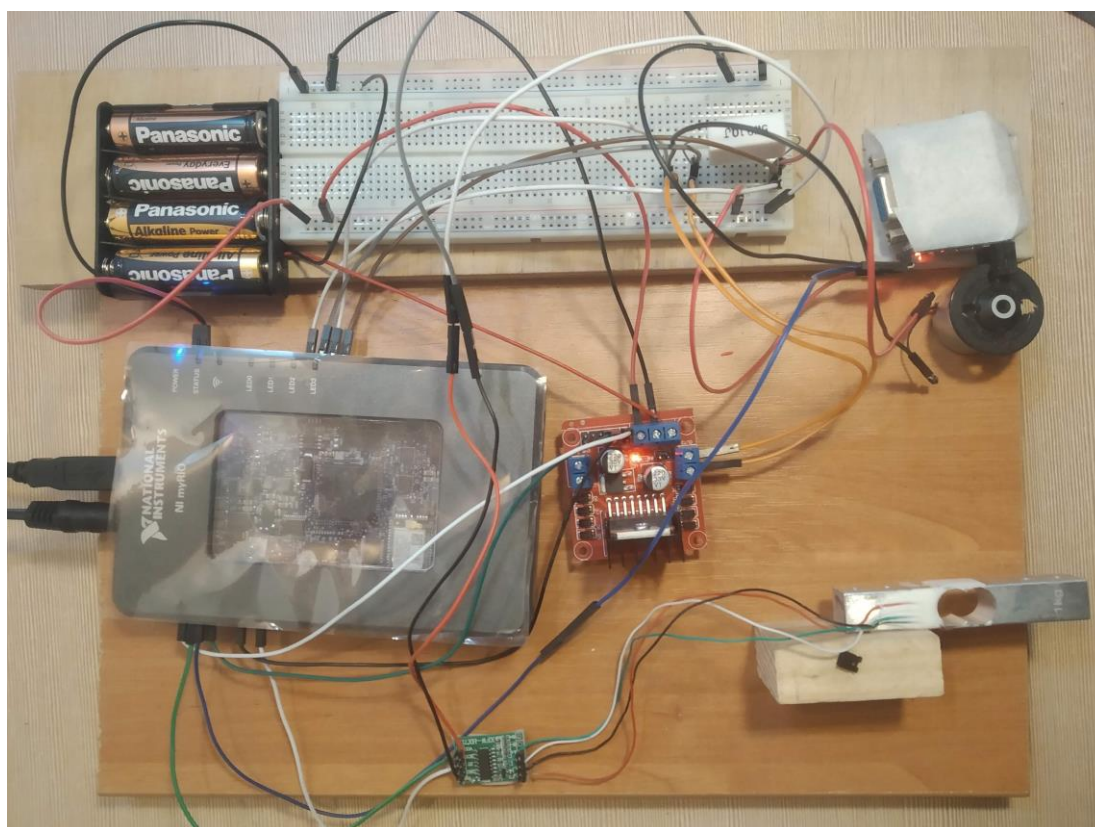
Rys. 14. Belka tensometryczna NA27. [10]



Rys. 15. Układ HX711. [11]



Rys. 16. Montaż silnika, belki tensometrycznej i czujnika przerwań oraz sposób pomiaru momentu obrotowego.



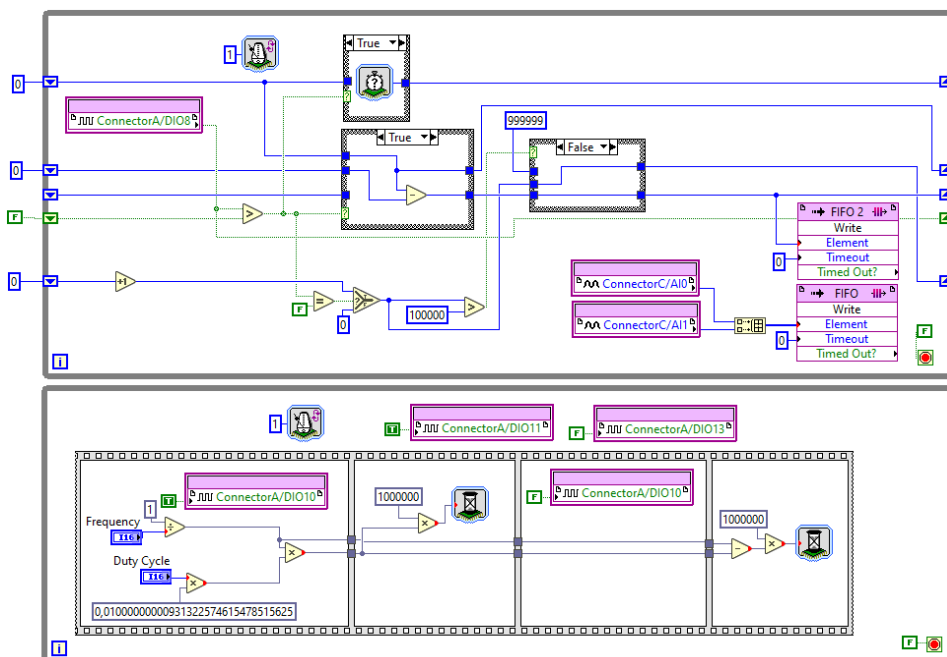
Rys. 17. Całe stanowisko pomiarowe.

4.2 Oprogramowanie stanowiska

Stworzono dwa programy do obsługi stanowiska kontrolno-pomiarowego w środowisku LabVIEW. Główny program dedykowany jest do przeprowadzania testów przez użytkownika podczas normalnej pracy silnika bez obciążenia. Mierzy on chwilową prędkość obrotową, natężenie prądu oraz napięcie. Drugi program został przygotowany do jednorazowego pomiaru momentu obrotowego oraz prądu, aby umożliwić wyznaczenie parametrów silnika.

Główny program podzielono na 3 moduły. Do kontroli silnika i zbierania informacji o jego działaniu wykorzystano moduł FPGA zapewniający bardzo szybkie działanie, gdyż poprawne przekazywanie i odbieranie sygnałów jest w aplikacji krytyczne. Moduł ten składa się z dwóch pętli While. Pierwsza pętla odpowiada za zbieranie informacji o działaniu silnika. Z każdą iteracją trwającą ok. 8 mikrosekund pobierana jest próbka sygnału cyfrowego z czujnika przerwań. Jeżeli wykryte jest zbocze rosnące (tj. w poprzedniej iteracji 0 a w aktualnej 1) to znaczy, że pomiędzy ramionami czujnika przerwań pojawił się otwór nakładki silnika. Czas tego zajścia zostaje zapisany funkcją Tick Count i jest przechowywany w rejestrze przesuwym, aż do wystąpienia kolejnego zbocza rosnącego. Czasy występowania następujących po sobie zbocz rosnących zostają odjęte i w ten sposób otrzymywana jest liczba mikrosekund jaka mija między wykryciami otworów silnika. Jako że, na nakładce silnika wykrywalny jest tylko jeden otwór, to obliczony czas jest okresem jednego obrotu silnika. Wynik ten zostaje przechowany w rejestrze przesuwym i wysyłany do kolejki FIFO w każdej iteracji pętli. Gdy zostaje wyznaczony kolejny okres obrotu nadpisuje on poprzedni. Aby zabezpieczyć program przed zwracaniem niepoprawnych wartości okresu w przypadku nagłego zatrzymania silnika (np. przytrzymanie go ręką) użyto dodatkowego rejestru przesuwego. Jego przetrzymywana wartość zwiększana jest o 1 w każdej iteracji i zerowana jeżeli wykryte jest zbocze rosnące z czujnika przerwań. W momencie, gdy licznik ten osiągnie wartość 100000 czyli ok.800 milisekund bez otrzymania kolejnych zbocz rejestr przesuwny z wartością okresu jest aktualizowany wartością informującą o zatrzymaniu silnika. W każdej iteracji są także zbierane wartości napięć na rezystorze oraz na zaciskach silnika. Są one wysyłane do drugiej kolejki FIFO.

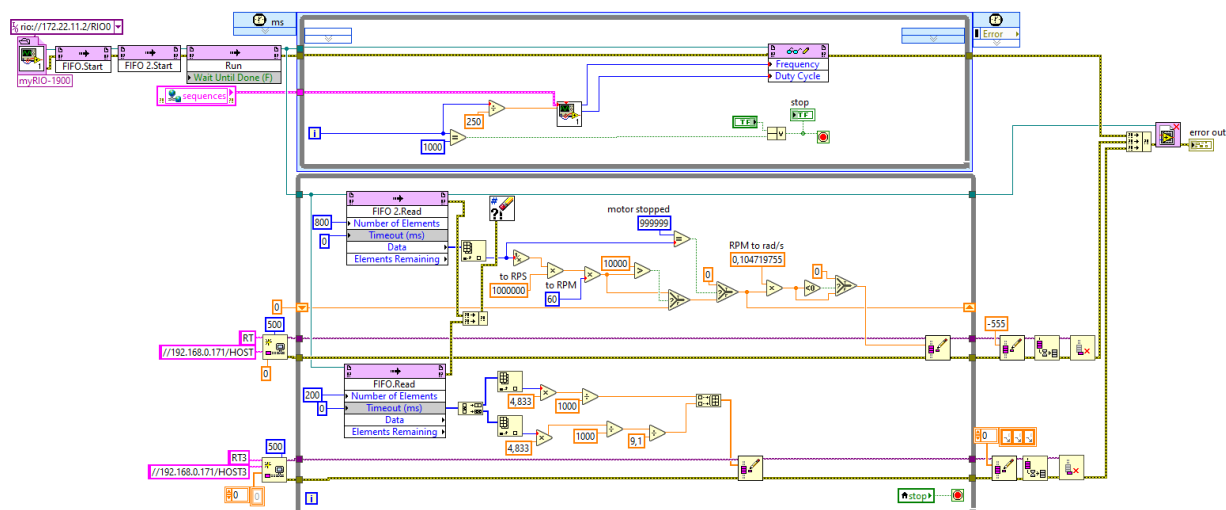
Druga pętla odpowiada za sterowanie silnikiem, za pomocą struktury Flat Sequence zaimplementowane jest generowanie sygnału PWM o wyznaczonej częstotliwości i wypełnieniu. Iteracja pętli trwa tyle co długość okresu sygnału. Na początku wystawiany jest stan wysoki, pętla czeka do momentu aż minie czas ustalony przez wypełnienie PWM a następnie wystawia stan niski i czeka czas jaki pozostał do końca trwania okresu.



Rys. 18. Moduł FPGA głównego programu.

Modułem odpowiedzialnym za odpowiednie przetworzenie danych jest RT, podobnie jak FPGA jest on uruchamiany na myRIO, ale jego działanie nie musi być aż tak szybkie. Na początku działania program uruchamia moduł FPGA. Na RT składają się dwie pętle. Pierwsza to pętla Timed Loop, jej działanie różni się od zwykłej pętli While tym, że ma ona określony okres, w którym musi się wykonać. Używa jej się do zadań, które muszą być wykonane w jakimś określonym czasie. W tym wypadku jest to wysyłanie do FPGA wartości częstotliwości i wypełnienia sygnału PWM. Na początku działania programu, odczytane zostają informacje o sekwencji obrotów silnika wybrane przez użytkownika i z pomocą SubVI'a „sequence.vi” na podstawie czasu jednej iteracji oraz ich licznika do FPGA wysłane zostają dane.

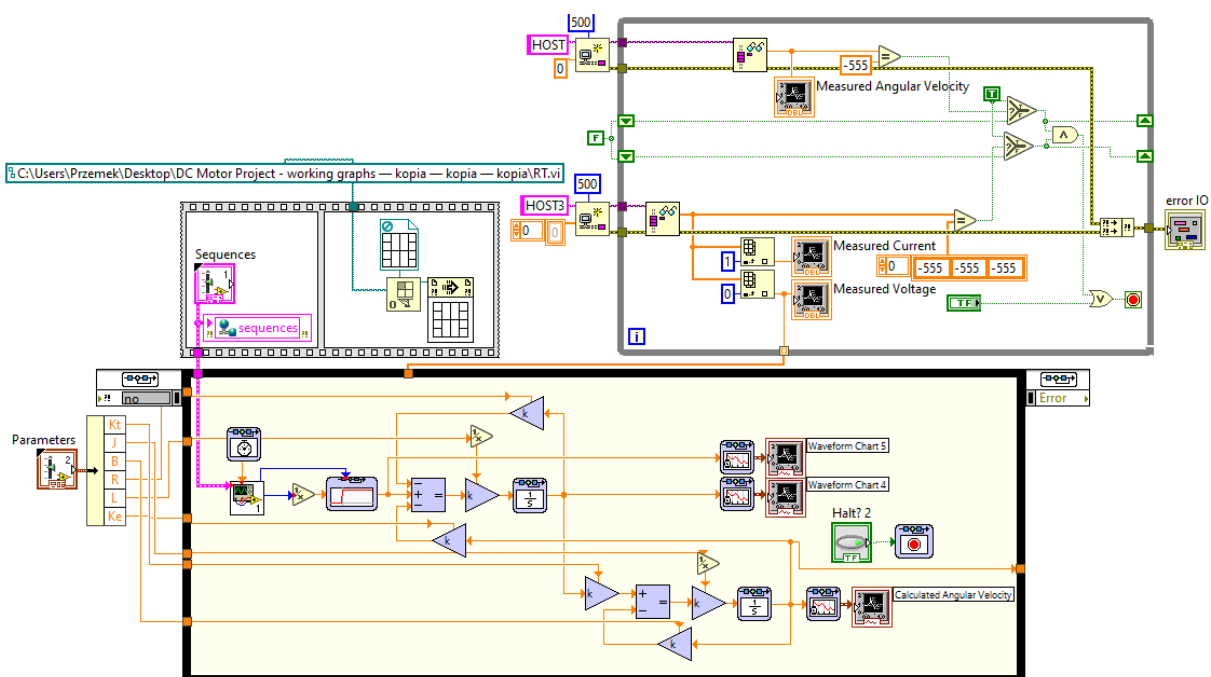
Druga pętla składająca się na moduł RT to pętla While. Przed jej startem zostaje nawiązane połączenie z modułem HOST przez Network Streams. W pętli tej z kolejek FIFO pomiary silnika zebrane w FPGA są odczytywane i odpowiednio przetwarzane. Okres obrotu silnika jest zamieniany na prędkość obrotową i uwzględnione jest odczytywanie informacji o nagłym zatrzymaniu silnika. Napięcie na rezystorze jest dzielone przez jego rezystancję, aby otrzymać wartość natężenia prądu przepływającego przez silnik. Obrobione informacje są przesyłane przez Network Streams do modułu HOST. RT kończy działanie, kiedy pętla Timed Loop wykona określoną liczbę iteracji, która odpowiada całkowitemu czasowi sekwencji. Zamykana jest komunikacja Network Streams i referencja modułu FPGA.



Rys. 19. Moduł RT głównego programu.

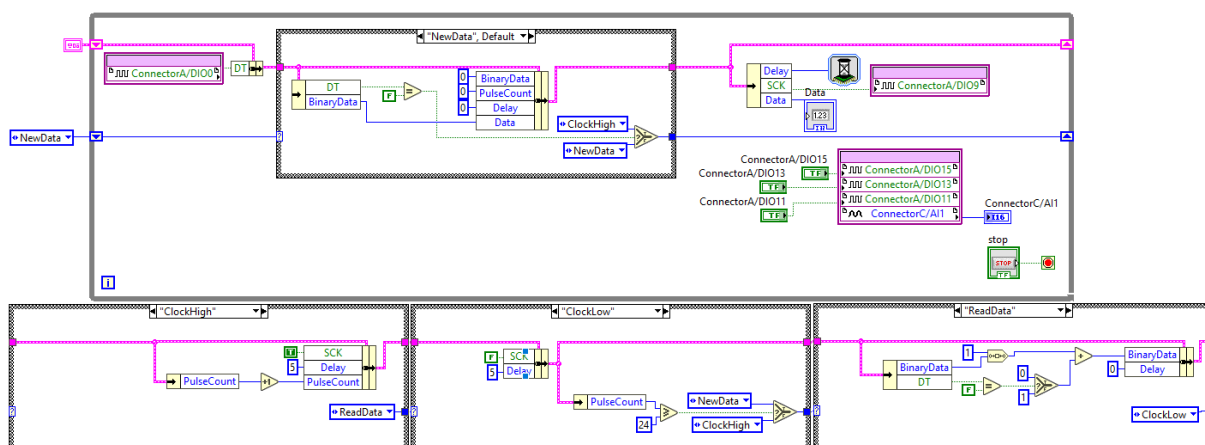
Moduł HOST służy do wybierania parametrów sekwencji i umożliwia obserwowanie jej wyników. Na panelu frontowym użytkownik wybiera długość trwania, częstotliwość i wypełnienie pięciu następujących po sobie sygnałów PWM sterujących silnikiem jak i parametry modelu matematycznego silnika. Wykresy służą do wyświetlania prędkości obrotowej, natężenia prądu jak i napięcia w czasie działania silnika. Są dwa źródła tych przebiegów. Jednym jest stanowisko pomiarowe i dane z czujników przy pracy silnika, a drugim model matematyczny i przeprowadzone za pomocą niego obliczenia.

Na schemacie blokowym na początku następuje zapisanie wartości parametrów sekwencji, a następnie uruchomiony zostaje moduł RT. Po zainicjalizowaniu komunikacji przez Network Streams w pętli While odbierane są z RT i wyświetlane na wykresach wyniki pomiarów. W tym module także jest zaimplementowany model matematyczny silnika, który korzysta z tych samych parametrów sekwencji oraz z parametrów silnika wybranych przez użytkownika.



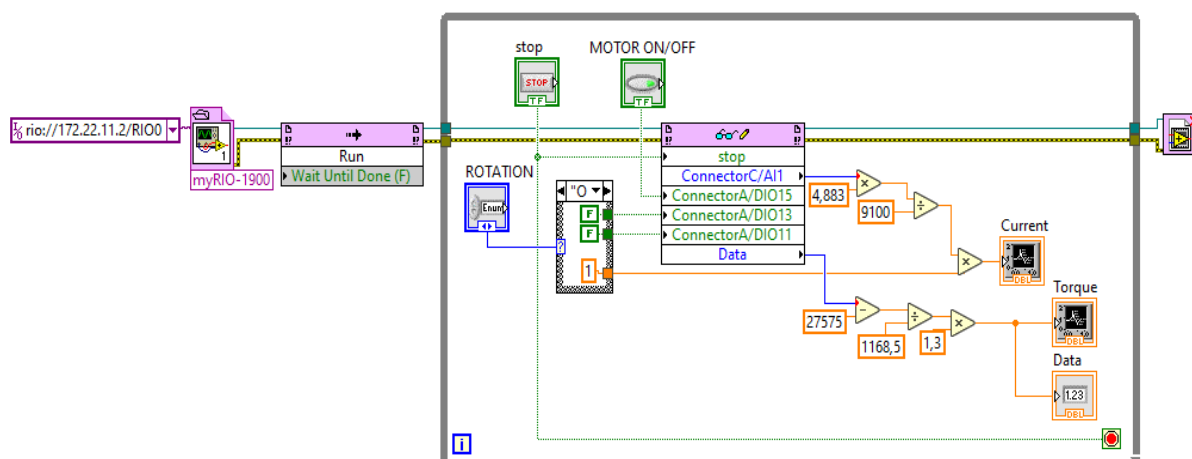
Rys. 20. Moduł HOST głównego programu.

Drugi program składa się z modułów RT i FPGA. Moduł FPGA podaje na wejście zegarowe przetwornika sygnał prostokątny o częstotliwości 100kHz i w każdym okresie czyta jeden bit danych z HX711 i dopisuje go do dotychczas zebranych jako najmłodszy bit. Po zebraniu 24 bitów zamienia je na wartość dziesiętną i rozpoczyna pobieranie następnych. W module tym zaimplementowano też włączanie/wyłączanie silnika i odczyt napięcia na rezystorze.



Rys. 21. Moduł FPGA programu do pomiaru momentu obrotowego.

Moduł RT przygotowano, aby odczytywać dane z tensometru oraz wartości prądu wyznaczone z napięcia na rezystorze zebrane przez FPGA. Belkę zamontowano najpierw pionowo jako wagę i skalibrowano ją przy użyciu przedmiotów o znanej wadze. Następnie umieszczono ją poziomo i dostosowano do pomiaru momentu obrotowego silnika z uwzględnieniem tego w jakiej odległości od wirnika montowana jest nitka łącząca belkę z silnikiem.



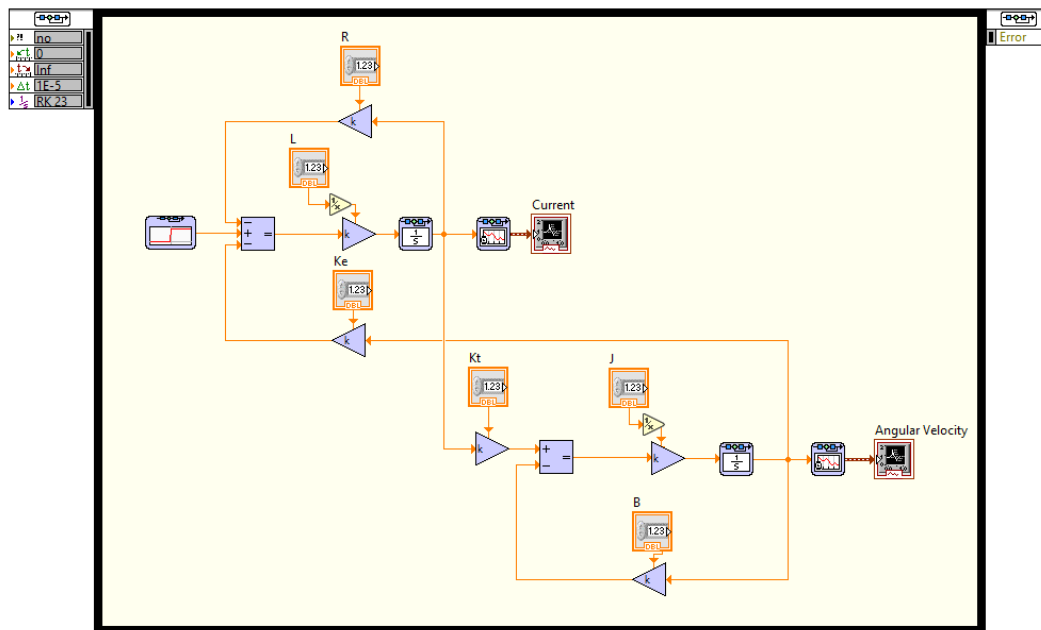
Rys. 22. Moduł RT programu do pomiaru momentu obrotowego.

4.3 Implementacja modelu matematycznego silnika

Układ równań opisujący działanie modelu matematycznego silnika przekształcono i na jego podstawie zaimplementowano schemat blokowy w środowisku LabVIEW przy użyciu Control & Simulation Loop.

$$\begin{cases} V_{in} = i * R + L * \frac{di}{dt} + K_e * \omega \\ K_t * i = J * \frac{d\omega}{dt} + B * \omega \end{cases}$$

$$\begin{cases} \frac{di}{dt} = -i * \frac{R}{L} - \frac{K_e}{L} * \omega + \frac{V_{in}}{L} \\ \frac{d\omega}{dt} = -\frac{B}{J} * \omega + \frac{K_t}{J} * i \end{cases}$$



Rys. 23. Schemat blokowy modelu matematycznego silnika zaimplementowany w środowisku LabVIEW.

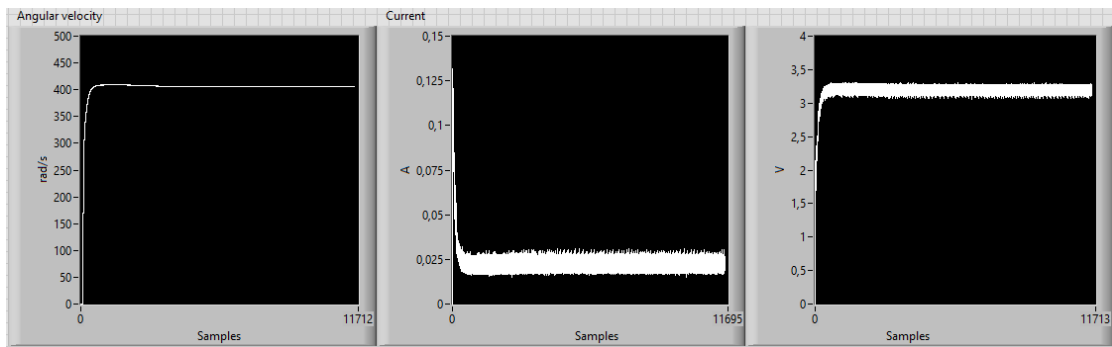
W modelu znajduje się 6 parametrów, które determinują jego działanie: rezystancja, indukcyjność, stała momentu obrotowego, stała elektromotoryczna, stała współczynnika tarcia oraz siła bezwładności. Aby móc odpowiednio porównać działanie wybranego silnika RF-300FA z jego modelem matematycznym, należy te parametry dobrać w oparciu o przeprowadzone na nim pomiary. Rezystancję zmierzono multimetrem i otrzymano wynik 9.8Ω . Indukcyjność została zmierzona przy pomocy oscyloskopu i generatora przebiegów, wykorzystując stałą czasową RL układu. Do generatora, na którym wystawiono sygnał prostokątny podłączono silnik z podpiętym szeregowo rezystorem 68Ω . Sondę oscyloskopu wpięto przed rezystorem i zmierzono czas jaki upływa od maksymalnej amplitudy sygnału do momentu, aż spadnie ona do 37% jej największej wartości. Zmierzony czas wyniósł 60 mikrosekund i była to stała czasowa RL tego układu. Wiedząc, że $\tau = L/R$, a całkowita rezystancja to $9,8 + 68 \Omega$, obliczono indukcyjność silnika, która wyniosła $0,004668 H$. Kiedy silnik pracuje w stabilnym stanie, ze stałą prędkością obrotową, a co za tym idzie ze stałym prądem, to wzór:

$$V_{in} = i * R + L * \frac{di}{dt} + K_e * \omega$$

można uprościć do:

$$V_{in} = i * R + K_e * \omega$$

i wyznaczyć z niego stałą elektromotoryczną. Wykorzystano do tego przygotowane już stanowisko i przeprowadzono pomiary.



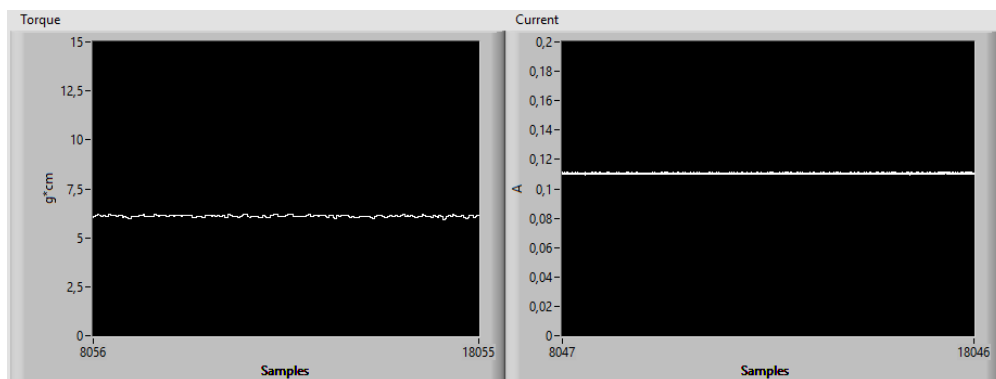
Rys. 24. Wykresy prędkości obrotowej, natężenia prądu i napięcia podczas pracy silnika bez obciążenia.

Uśrednione wartości przebiegów to:

- $V_{in} = 3.19V$
- $i = 0,023A$
- $\omega = 406 \text{ rad/s}$

Podstawiając wartości do wzoru otrzymujemy $K_e = 0,0073$

K_t obliczono ze wzoru $T = K_t * i$ przy użyciu belki tensometrycznej



Rys. 25. Wykresy momentu obrotowego i natężenia prądu zatrzymanego silnika.

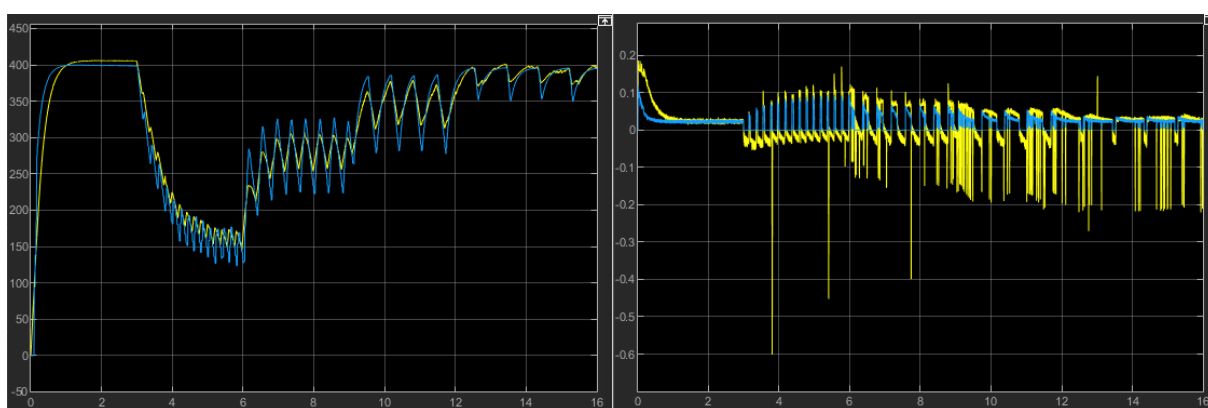
Średnia wartość momentu obrotowego wyniosła $6,075 \text{ g} * \text{cm}$ natężenia prądu przytrzymywanego silnika $0,1103 \text{ A}$. Podstawiając do wzoru, wyznaczone K_t wynosi $0,0053$. Analogicznie jak poprzednio możemy także przekształcić drugi wzór:

$$K_t * i = J * \frac{d\omega}{dt} + B * \omega$$

$$K_t * i = B * \omega$$

Wyznaczone z niego B wynosi $3 * 10^{-7}$

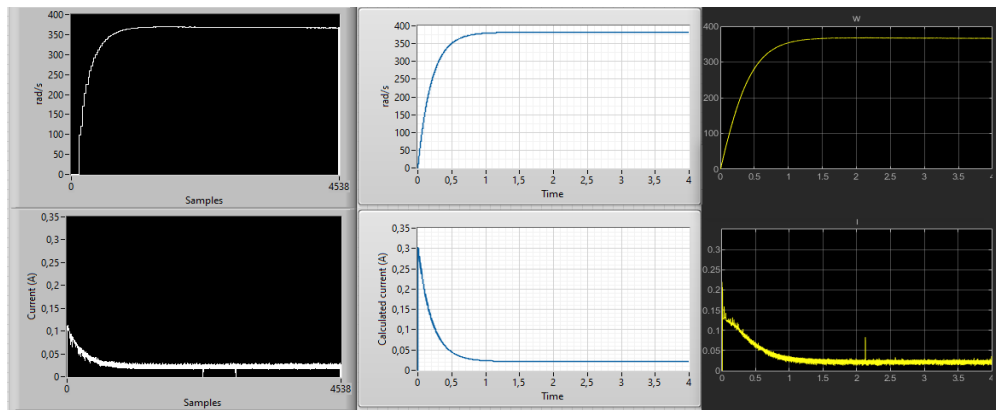
Do wyznaczenia ostatniego parametru posłużono się programem Simulink. Sporządzono diagram blokowy modelu silnika, a na jego wejście podano odczytane wartości napięcia z przeprowadzonej wcześniej sekwencji. Przy pomocy narzędzia Parameter Estimation, które na podstawie przebiegów wyjściowych szacuje parametry modelu wyznaczono $J = 8,5 * 10^{-7}$. Po przeprowadzeniu symulacji modelu w Simulinku z wszystkimi wcześniej wyznaczonymi parametrami i przy rzeczywistym napięciu sterującym z myRIO uzyskano pokazane niżej przebiegi prędkości obrotowej oraz natężenia prądu. Niebieskie wykresy to przebiegi zmierzone przy pomocy stanowiska pomiarowego, a żółte to te obliczone przez model. Prądy poniżej zera nie występują na wykresie zmierzonego prądu ponieważ mostek H nie pozwala na jego przepływ w drugą stronę. Poza tym wykresy są do siebie bardzo zbliżone.



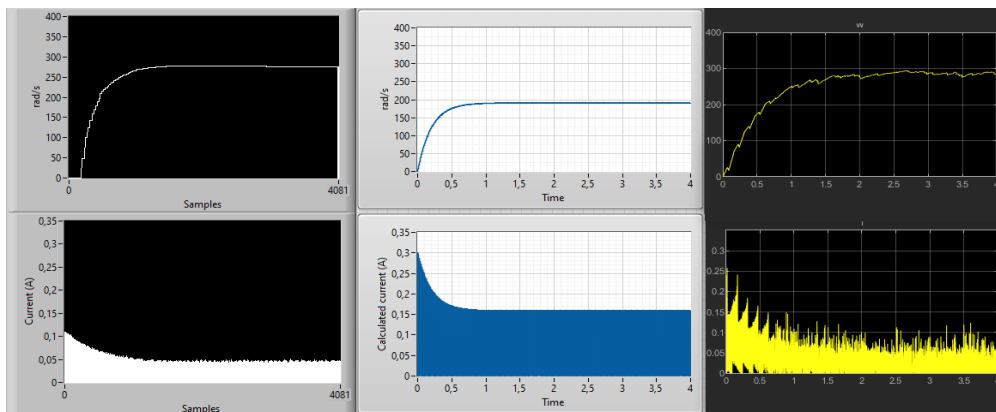
Rys. 26. Wykresy prędkości obrotowej i natężenia prądu wyznaczone za pomocą stanowiska pomiarowego oraz przez model bazując na rzeczywistym napięciu sterującym.

4.4 Przeprowadzenie testów

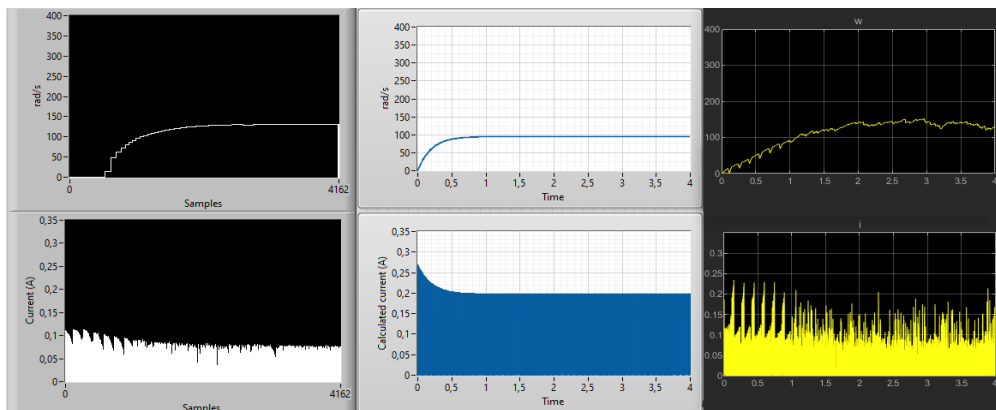
Przeprowadzono testy porównując przebiegi prędkości obrotowej i natężenia prądu w trzech przypadkach. Wykresy po lewej stronie (A) obrazują dane dotyczące rzeczywistego silnika odczytane za pomocą stanowiska kontrolno-pomiarowego. Środkowe wykresy (B) przedstawiają działanie modelu silnika o wyznaczonych parametrach i o idealnym sygnale sterującym. Wykresy po prawej stronie (C) przygotowano w programie Simulink, pokazują one działanie modelu silnika o wyznaczonych parametrach, ale z rzeczywistym sygnałem sterującym zmierzonym przy pomocy stanowiska kontrolno – pomiarowego. Częstotliwość sygnału PWM przy wszystkich pomiarach została ustawiona na 250 Hz.



Rys. 27. Wypełnienie PWM 100%.



Rys. 28. Wypełnienie PWM 50%.



Rys. 29. Wypełnienie PWM 25%.

Między wykresami A i B występują spore odstępstwa, szczególnie przy wypełnieniu 50% oraz na wykresie prądu przy wypełnieniu 25%. Wykresy C pokazują, że tak samo sparametryzowany model z rzeczywistym sygnałem sterującym mimo postrzępionego przebiegu osiąga podobne wartości do tych na wykresach A. Różnice te prawdopodobnie wynikają z nieidealnego sygnału sterującego silnikiem oraz niedokładności pomiarów bardzo szybko zmieniającego się napięcia.

V. Podsumowanie

Założenia pracy zostały spełnione. W ramach projektu zbudowano stanowisko kontrolno-pomiarowe oraz oprogramowanie do jego obsługi oparte o graficzne środowisko programowania – LabVIEW. Ponadto, pomyślnie wyznaczono parametry silnika i zaimplementowano jego model.

Zbudowane stanowisko kontrolno-pomiarowe pozwoliło na pomiar momentu obrotowego silnika w oparciu o belkę tensometryczną, a także jego prędkości obrotowej mierzonej z wykorzystaniem czujnika podczerwieni. Akwizycja danych zaimplementowana została w układzie FPGA kontrolera. W systemie oprogramowano układ RT, umożliwiając transfer danych z FPGA do urządzenia typu PC. W stworzonej pod Windows OS aplikacji porównano osiągi rzeczywistego silnika z jego matematycznym modelem, a przy pomocy programu Simulink wyznaczono dodatkowo przebiegi osiągow teoretycznego modelu silnika z rzeczywistym sygnałem sterującym. Pomogło to wyjaśnić skąd mogą brać się niespójności w działaniu obiektów.

Korzystanie z narzędzia jakim jest model matematyczny jest bardzo pomocne przy projektowaniu wszelkich układów, trzeba jednak zawsze brać pod uwagę, że rzeczywisty układ nie będzie osiągał takich samych wyników jak jego teoretyczne odwzorowanie z uwagi na nieidealność komponentów i danych wejściowych czy czynniki zewnętrzne.

VI.Bibliografia

- [1] Irving Gottlieb, „Practical Electric Motor Handbook, ” 1997
- [2] Jacek Przepiórkowski, „Silniki elektryczne w praktyce elektronika,” 2007
- [3] Xiaocong Fan,,„Real-Time Embedded Systems: Design Principles and Engineering Practices,” 2015.
- [4] Pong Chu, „FPGA prototyping by Verilog examples,” 2008.
- [5] National Instruments, „myRIO-1900 User Guide and Specifications, ”
- [6] <https://product.mabuchi-motor.com/detail.html?id=74> [data dostępu: 05.01.2020]
- [7] https://c1.neweggimages.com/ProductImage/AHBU_1_201901191174762662.jpg [data dostępu: 05.01.2020]
- [8] https://cdn.shopify.com/s/files/1/0069/6513/3376/products/AA034_2.jpg?v=1547714206 [data dostępu: 05.01.2020]
- [9] https://obrazki.elektroda.pl/9434569400_1540541669.jpeg [data dostępu: 05.01.2020]
- [10] https://static5.abc-rc.pl/pol_pl_Belka-tensometryczna-NA27-1kg-10N-YZC-131-7317_1.jpg [data dostępu: 05.01.2020]
- [11] https://static2.abc-rc.pl/pol_pl_Modul-miernik-nacisku-HX711-modul-wagi-tensometru-ARDUINO-6860_1.png [data dostępu: 05.01.2020]