



na kierunku Mechatronika

w specjalności Mechatronika Pojazdów

Gear.py - kalkulator przekładni walcowych. Implementacja w języku Python

numer pracy według wydziałowej ewidencji prac {liczba}

Mateusz Szawczenko

Numer albumu 282619

promotor

dr inż. Krzysztof Twardoch

konsultacje

-----

Warszawa 2021

Mateusz Szawczenko

**OŚWIADCZENIE**

Jako autor pracy dyplomowej pt.: „Gear.py - kalkulator przekładni walcowych, implementacja w języku Python”, którą wykonałem samodzielnie przestrzegając zasad ochrony własności intelektualnej zezwalam na publiczne udostępnienie pracy i wyrażam zgodę na jej udostępnienie w Bibliotece Wydziału Samochodów i Maszyn Roboczych Politechniki Warszawskiej w ramach realizacji zadań statutowych biblioteki.

Warszawa, dnia ……………………… ……………………………….

/podpis/

**SŁOWA KLUCZOWE**

1. Aplikacja komputerowa

2. Przekładnie walcowe

3. Kalkulator przekładni

4. Python

Streszczenie:

Głównym celem niniejszej pracy inżynierskiej było stworzenie interfejsu graficznego oraz zaimplementowanie do niego algorytmu obliczeń parametrów przekładni walcowych, tak aby umożliwić użytkownikowi projektowanie przekładni w jasny i przejrzysty sposób.

Zadaniem użytkownika jest wprowadzenie danych do programu i głównych założeń, a program wykonuje wszystkie potrzebne obliczenia i zwraca gotowe rozwiązanie wraz z uproszczonym rysunkiem schematu przekładni.

Do stworzenia interfejsu graficznego zastosowany został język programowania ‘Python’ ze względu na wiele dostępnych bibliotek ułatwiających tworzenie i programowanie elementów graficznych oraz możliwość implementacji w nim wcześniej przygotowanych obliczeń.

Sam interfejs został stworzony przy pomocy narzędzia QTDesigner, które umożliwia projektowanie przy użyciu gotowych, zwizualizowanych elementów takich jak np. ramki, przyciski, itp. Ponadto, można je dostosowywać pod względem wielkości, położenia, kolorystyki w dowolny sposób. Efektem końcowym jest wygenerowany kod w języku Python do którego trzeba dopisać reakcje na zdarzenia, czyli np. wykonanie odpowiedniej funkcji po wciśnięciu przycisku przez użytkownika programu.

Największym problemem było przełożenie obliczeń z języka Fortran do Python’a, co zostało rozwiązane z użyciem gotowej biblioteki F2Py, stworzonej specjalnie do tego celu. Umożliwiło to połączenie interakcji użytkownika z algorytmem poprzez definiowanie odpowiednich danych wejściowych dla algorytmu obliczeniowego i odpowiednie wyświetlenie ich w interfejsie graficznym.

Summary:

Main purpose of this final engineering work was to create graphical user interface and to implement bevel gear’s computational algorithm to it. It allows user to design bevel gears in a clear and transparent way.

The user has to only put input data to the program and some main assumptions. Program does all required calculations and returns the solution together with simplified drawing of bevel gear scheme.

Python programming language has been used to create the graphical interface due to its many available, work facilitating libraries to create and program graphical elements. It also offers possibility of implementing to it previously prepared calculations in Fortran language.

The graphical user interface has been made using QTDesigner tool, which allows designing by using ready-made, visualized elements such as frames, buttons, labels, etc. Furthermore, the elements can be adjusted in terms of their size, location, color in any way. As the final result this tool returns code in Python to which I only had to add some user driven events reactions, that is, for example execution of proper function when user presses some buttons.

The biggest issue was to translate Fortran’s calculations to Python language, which has been resolved thanks to F2Py library created specially, for this purpose. It made possible, to connect user’s interaction with the computational algorithm by defining appropriate input data to the algorithm and their suitable use and display in the graphical user interface.

Spis treści

[1. Wstęp 6](#_Toc72023259)

[2. Przegląd rozwiązań komputerowych do obliczania przekładni walcowych 7](#_Toc72023260)

[2.1 MITCalc 7](#_Toc72023261)

[2.2 PRZEKŁADNIA 8](#_Toc72023262)

[3. Biblioteki, rozszerzenia i technologie informatyczne wykorzystywane w programie Gear.py 8](#_Toc72023263)

[3.1 QtDesigner[1] 8](#_Toc72023264)

[3.2 PyQt5[2] 9](#_Toc72023265)

[3.3 CSS[3] 10](#_Toc72023266)

[1:25:00 12](#_Toc72023267)

[Sys 12](#_Toc72023268)

[Pandas 12](#_Toc72023269)

[Numpy 12](#_Toc72023270)

[4. Główne założenia programistyczne 13](#_Toc72023271)

[5. Składowe aplikacji 13](#_Toc72023272)

[6. Interfejs graficzny 13](#_Toc72023273)

[6.1 Start 13](#_Toc72023274)

[6.2 Dane 13](#_Toc72023275)

[6.3 Kształt 13](#_Toc72023276)

[6.4 Raport 13](#_Toc72023277)

[7. Podstawy teoretyczne obliczeń [4] 14](#_Toc72023278)

[Dobór materiału 19](#_Toc72023279)

[8. Implementacja algorytmu obliczeniowego 20](#_Toc72023280)

[9. Pozostałe funkcje 20](#_Toc72023281)

[10. Weryfikacja poprawności obliczeń na przykładzie 20](#_Toc72023282)

[11. Podsumowanie, wnioski i uwagi końcowe 20](#_Toc72023283)

[12. Bibliografia 20](#_Toc72023284)

[13. Spis tabel i rysunków 20](#_Toc72023285)

# Wstęp

Otaczający nas świat nieustannie się zmienia. Wiele zagadnień poruszanych przez wcześniejsze pokolenia inżynierów wygląda obecnie zupełnie inaczej. To wszystko za sprawą rozwoju technik informatycznych i ciągle rosnącej mocy obliczeniowej komputerów. Obliczenia, które kiedyś wymagały ogromnej wiedzy i wielu godzin pracy, dzisiaj można wykonywać za pomocą kliknięcia, a zakres wiedzy potrzebnej do rozwiązania niektórych problemów inżynieryjnych, sprowadza się często do umiejętności prawidłowej obsługi dedykowanej aplikacji komputerowej. Komputery i techniki informatyczne tak bardzo zmieniły pracę inżynierów na całym świecie, że obecnie chyba nikt z nas nie wyobraża sobie pracy bez nich. Niewątpliwą zaletą wykorzystywania komputerów do obliczeń jest możliwość wielokrotnego używania raz przygotowanych szablonów, schematów, algorytmów, bez konieczności zaczynania wszystkiego od początku, ilekroć w niewielkim stopniu zmienią się dane wejściowe, czy podstawowe założenia projektu. Uważam jednak, że jak do wszystkiego, tak i do wykorzystania komputerów w pracy inżynieryjnej należy podchodzić z pewną rezerwą zaufania. Jak każde inne narzędzie, jest ono na tyle dobre, na ile kompetentna jest osoba je obsługująca. Warto więc postrzegać moc obliczeniową komputerów jako ułatwienie, a nie zastąpienie pracy dzisiejszych inżynierów.

Celem niniejszej pracy, było stworzenie aplikacji ułatwiającej wykonywanie obliczeń parametrów przekładni walcowych, doboru odpowiednich średnich kół zębatych wynikających z obliczeń wytrzymałościowych, przy wprowadzanych przez użytkownika założeniach, takich jak: moment obciążający, prędkość kątowa zębnika, materiał z którego mają zostać wykonane koła i wiele innych. Aplikacja ma stanowić sposób weryfikacji zagadnień projektowych poruszanych na przedmiotach: Projektowanie Podstaw Konstrukcji Maszyn oraz Projektowanie Napędów Mechanicznych.

Do wykonania niniejszej pracy wykorzystany został język programowania Python i wiele wchodzących w jego skład bibliotek. Wszystkie aspekty projektowania aplikacji Gear.py zostały szczegółowo opisane w kolejnych rozdziałach, zarówno od strony procesu tworzenia szaty graficznej aplikacji, jak i metodyki zaimplementowanych w niej algorytmów obliczeniowych.

Mając na uwadze powyższe, obranych zostało kilka kluczowych założeń, które spełnia aplikacja Gear.py:

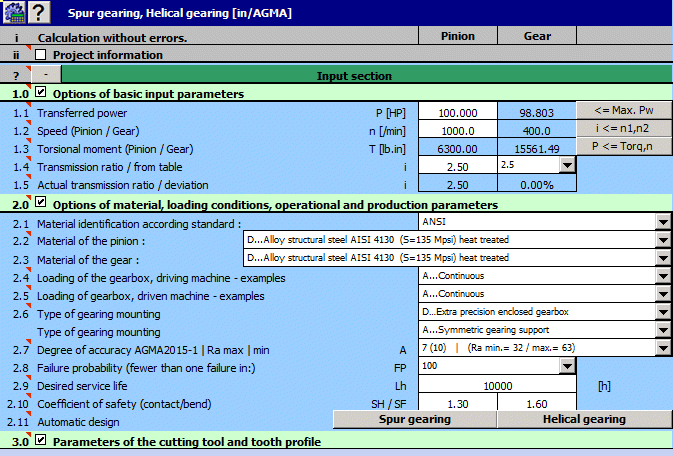
* Przejrzysty i łatwy w obsłudze (intuicyjny) interfejs graficzny – każde interaktywne pole, wymagające od użytkownika wprowadzenia danych zostało opisane wskazującą jego przeznaczenie nazwą. Ponadto za pomocą tzw. „hintów” (z ang. wskazówek), wyświetlających się po najechaniu kursorem, użytkownikowi wyświetla się bardziej szczegółowy opis pola.
* Wykorzystanie technik programowania obiektowego – umożliwia wielokrotne użycie raz napisanego kodu.
* Weryfikacja wprowadzanych przez użytkownika danych – na pola, w których konieczna jest interakcja z użytkownikiem nałożone zostały warunki uniemożliwiające wprowadzenie błędnych danych.
* Fragmentacja wykonywanych obliczeń – obliczenia podzielone zostały na kilka etapów, co ułatwia użytkownikowi weryfikację poprawności działania programu oraz pozwoli szybciej i łatwiej dostrzec ewentualne błędy w jego założeniach.
* Niskie wymagania sprzętowe.
* Brak konieczności instalowania dodatkowego oprogramowania w celu uruchomienia aplikacji.

# Przegląd rozwiązań komputerowych do obliczania przekładni walcowych

## MITCalc

**MITCalc** to złożone narzędzie oferujące zarówno pomoc w projektowaniu jak i sprawdzaniu obliczeń w wielu powszechnych dziedzinach inżynieryjnych, m. in.: przekładnie zębate o zazębieniu wewnętrznym i zewnętrznym (Rysunek 2.1), przekładnie stożkowe, przekładnie planetarne, przekładnie   
pasowe. Wadami programu są:

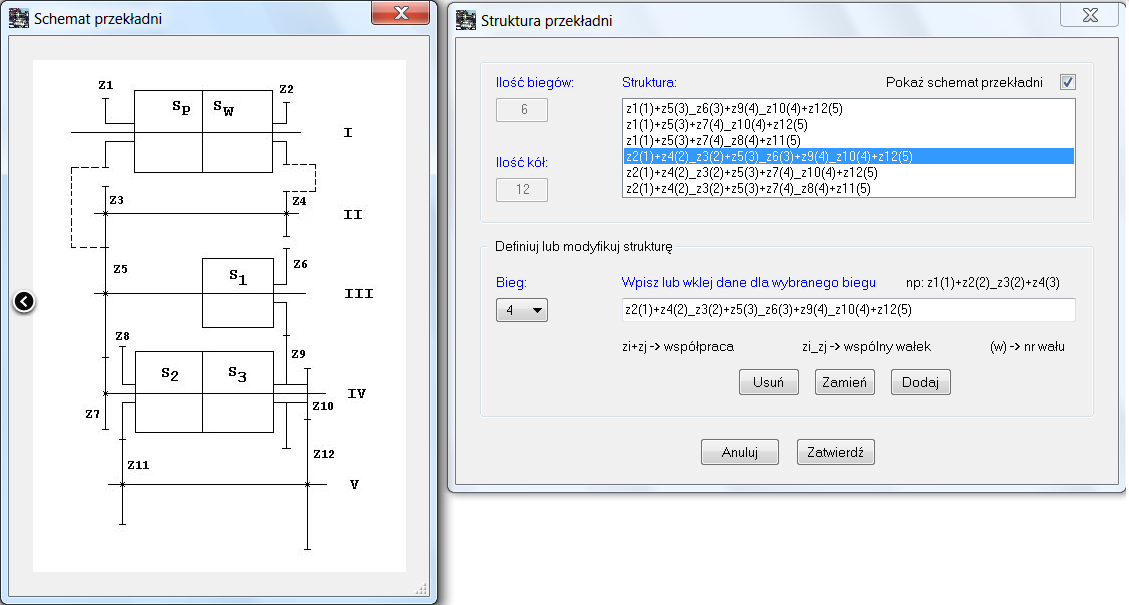
* Kompatybilność tylko z systemem Microsoft Windows;
* Przestarzały, mało intuicyjny interfejs;



Rysunek 1 MITCalc Spur Gear - interfejs użytkownika (<https://www.mitcalc.com/en/ui/ui_spurgear.htm>)

## PRZEKŁADNIA

Program komputerowy PRZEKŁADNIA, służy do projektowania przekładni zębatych o prostej linii zęba. Mogą to być przekładnie klasyczne, o różnym stopniu złożoności, jak również przekładnie typu ‘power shift’. Program ten pozwala na wielokryterialną optymalizację przekładni ze względu na różne, wybierane skojarzenia kryteriów. Obliczenia wytrzymałościowe przekładni można aktualnie realizować według normy ISO 6336.[[1]](#footnote-1)



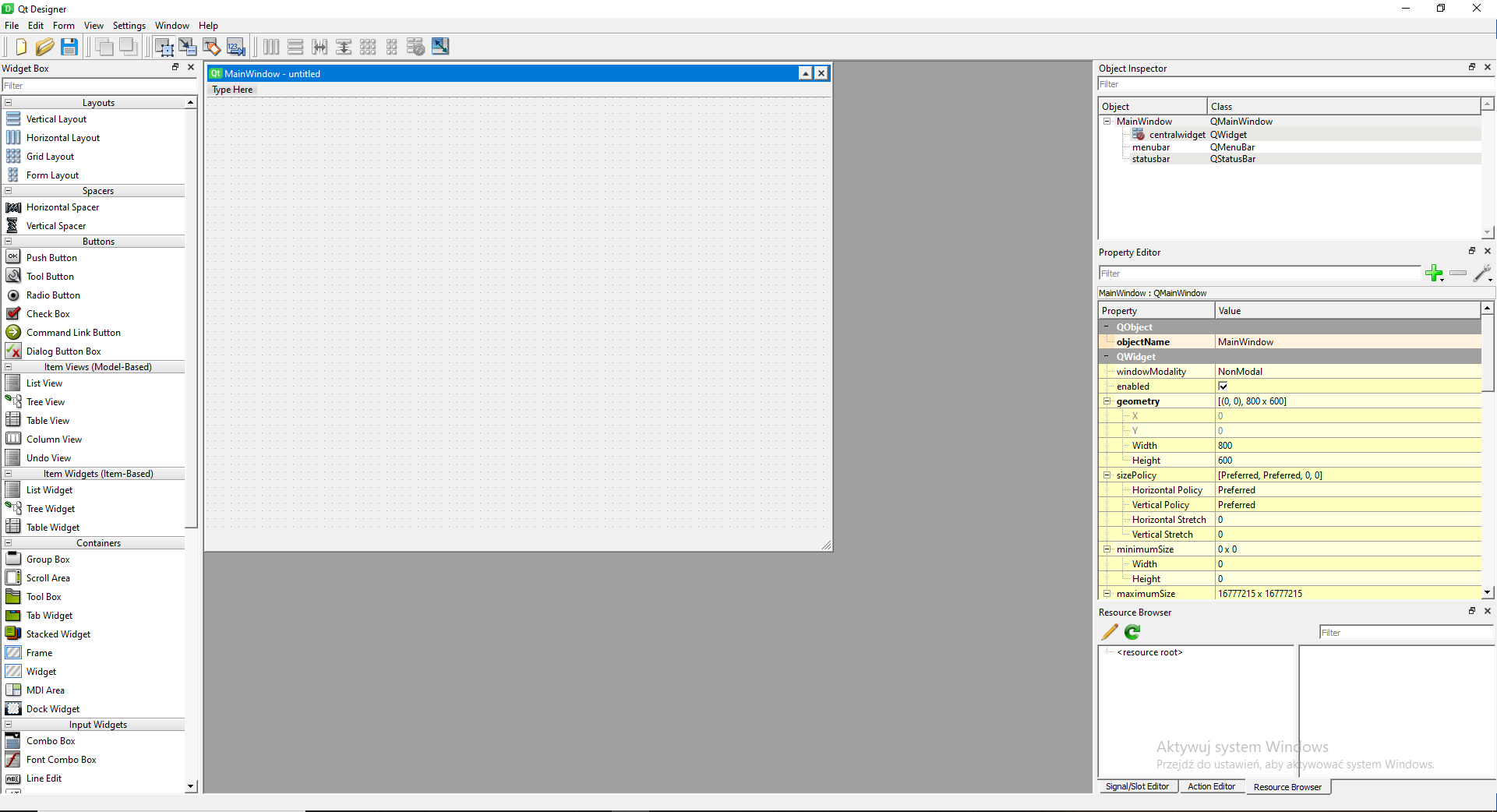
Rysunek 2 Schemat i struktura przekładni w programie PRZEKŁADNIA (<https://www.gearbox.com.pl/pl/zrzuty_ekranu.html>)

# Biblioteki, rozszerzenia i technologie informatyczne wykorzystywane w programie Gear.py

## QtDesigner[1]

**Qt Designer** to narzędzie stworzone z myślą o projektowaniu interfejsów graficznych przy użyciu tzw. „widgetów”. Umożliwia komponowanie i dostosowywanie wedle potrzeb wyglądu elementów składowych interfejsu w myśl zasady „what you see is what you get”, czyli wygląd końcowy w identyczny sposób odzwierciedla projektowany przez nas model.

Widgety i formuły stworzone za pomocą programu Qt Designer płynnie integrują się z kodem za pomocą mechanizmu sygnałów i slotów, dzięki czemu w łatwy sposób można przypisać oddziałujące z akcjami użytkownika zachowania poszczególnych elementów graficznych. Zmian można dokonywać zarówno w programie, jak i bezpośrednio w wygenerowanym przez niego kodzie. Narzędzie jest na tyle rozbudowane, że umożliwia również tworzenie i dodawanie swoich własnych, spersonalizowanych komponentów.



Rysunek 3 Ekran startowy programu Qt Designer

## PyQt5[2]

PyQt to jeden z najlepszych obecnie dostępnych, multi-platformowych zbiorów gotowych narzędzi do tworzenia interfejsów graficznych. Zapewnia pełną kontrolę nad elementami składającymi się na UI (z ang. „user intreface”), poprzez dowolną możliwość konfiguracji ich położenia, przynależności, rozmiarów czy też stylów kolorystycznych. Wspomniany we wcześniejszym podrozdziale Qt Designer tworzy kod źródłowy w oparciu właśnie o tę bibliotekę. Znajduje również zastosowanie w tworzeniu aplikacji mobilnych na systemy Android i iOS. Umożliwia rozwinięcie w pełni funkcjonalnych interfejsów, wraz z animacjami, czy nawet integracją z Google Maps.

Biblioteka PyQt5 jest kompatybilna z wieloma systemami operacyjnymi, takimi jak: Windows, Linux, UNIX, wspomniany już wcześniej Android czy też iOS. Do jego działania wymagana jest jedynie wersja 3.5 lub nowsza Python’a zainstalowana na urządzeniu. Biblioteka może również być osadzona w aplikacjach opartych na C++, co umożliwia rozszerzenie funkcjonalności takich aplikacji[[2]](#footnote-2).

W aplikacji Gear.py została wykorzystana pełnia możliwości tej biblioteki. Każdy element interfejsu został stworzony przy jej użyciu, począwszy od okna głównego programu, poprzez wszystkie okna dzielące ekran główny na sekcje, a skończywszy na wszystkich widocznych w aplikacji przyciskach, okienkach dialogowych czy też ramkach umożliwiających użytkownikowi wprowadzanie danych.

Poniżej przedstawiam fragment kodu wykorzystującego możliwości tej biblioteki, który posłużył do podziału górnej części głównego okna aplikacji na sekcje takie jak: main\_header (główny nagłówek), tittle\_bar\_container, left\_menu\_toggle. Dopiero po takim podziale górnej części okna aplikacji, w każdej z nich można było umieścić kolejne elementy, np. przycisk rozwijający menu w sekcji left\_menu\_toggle.



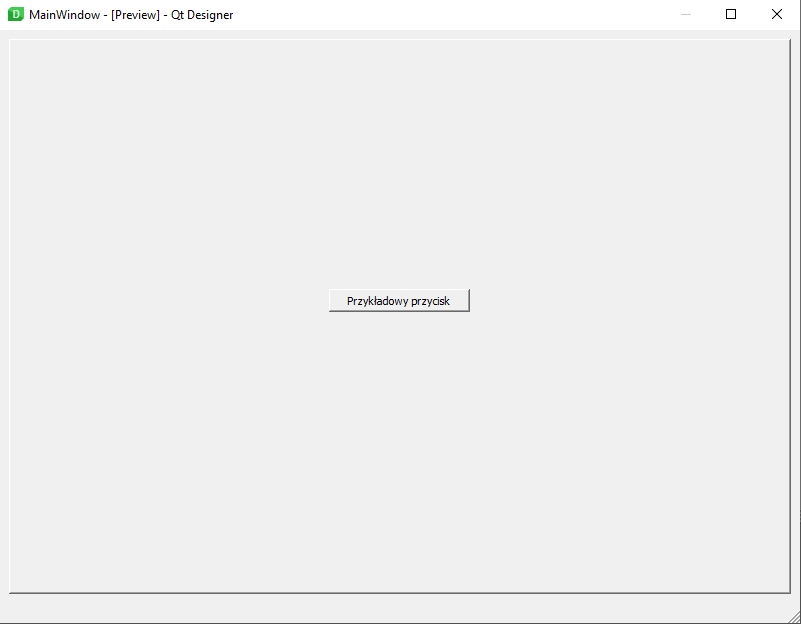
Rysunek 4 Fragment kodu aplikacji Gear.py, w którym wykorzystana została biblioteka PyQt5

## CSS[3]

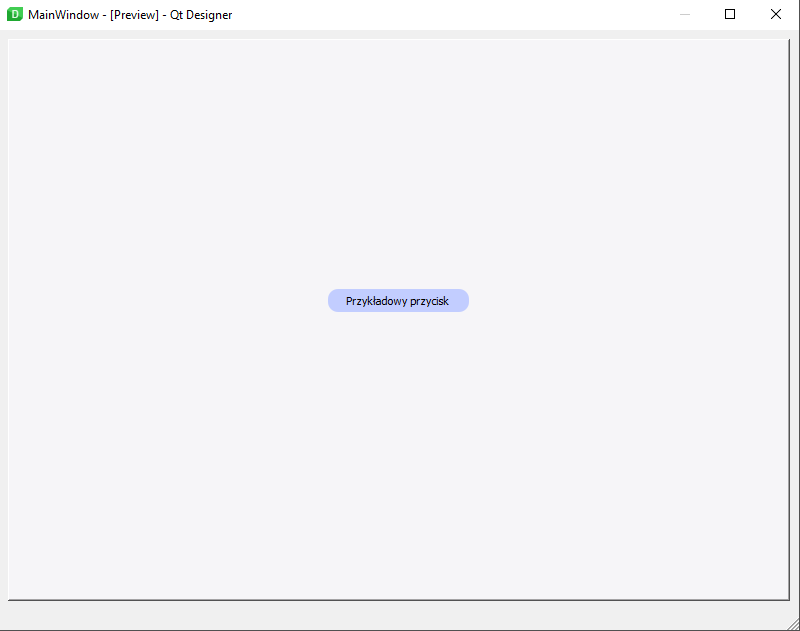
Myśląc o stworzeniu aplikacji Gear.py, jeszcze przed rozpoczęciem pracy nad nią, byłem pewien jednego – chciałbym, aby wyglądem było jej jak najbliżej do aplikacji robionych przez profesjonalnych twórców obecnie, aniżeli do tych, które znamy z lat 90. ubiegłego stulecia. Większość kursów, które dotyczyły tworzenia interfejsów graficznych, nawet te oparte o bibliotekę PyQt5 uczyło tworzenia interfejsów, które swoim wyglądem nie zaspokajały moich potrzeb. Rozwiązaniem mojego problemu okazał się właśnie CSS.

CSS (ang. Cascading Style Sheets), czyli kaskadowe arkusze stylów to kod, dzięki któremu mogłem dowolnie spersonalizować wygląd każdego elementu, który znalazł się w aplikacji Gear.py według mojej wizji i wcześniejszych założeń. Podobnie jak HTML, nie jest on uznawany za język programowania, jest po prostu językiem arkuszy stylów.

Poniżej przedstawiam porównanie przycisku stworzonego w bibliotece PyQt5, przed i po zastosowaniu CSS, celem spersonalizowania jego wyglądu, aby umożliwić czytelnikowi lepsze zrozumienie tego, jak ważny w mojej pracy okazał się być CSS.

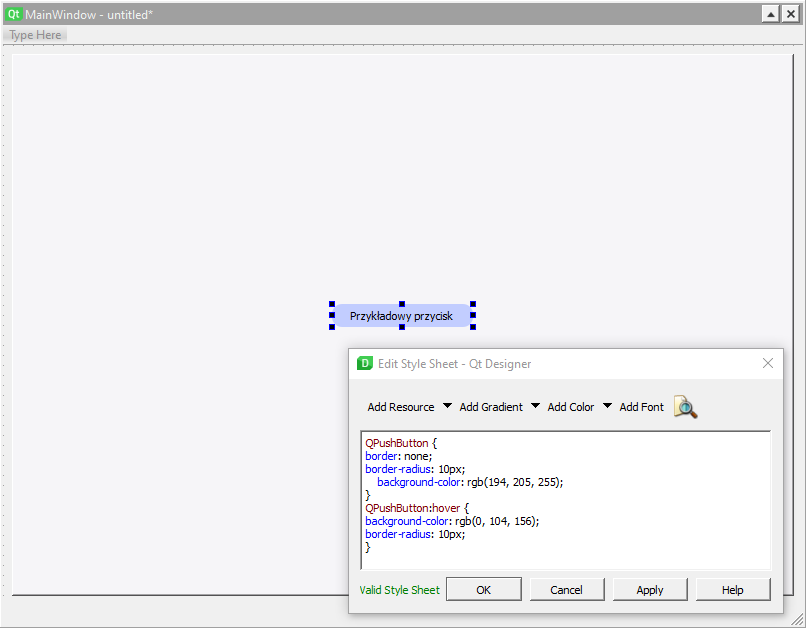


Rysunek 5 Przykład przycisku bez użycia CSS



Rysunek 6 Przykład przycisku po zastosowaniu CSS

Kod CSS i jego aplikacja w programie Qt Designer:



Rysunek 7 Modyfikacja przycisku w programie Qt Designer

Dzięki tym kilku linijkom kodu (Rysunek 7), mogłem zupełnie zmienić wygląd przycisku, a nawet nadać mu cechy, które uwidoczniają się tylko po najechaniu na niego kursorem myszy, dzięki funkcji „hover”.

## 1:25:00

## Sys

## Pandas

## Numpy

# Główne założenia programistyczne

# Składowe aplikacji

# Interfejs graficzny

## Start

## Dane

## Kształt

## Raport

# Podstawy teoretyczne obliczeń [4]

Głównym zadaniem przekładni mechanicznych jest przenoszenie energii z jednego punktu do drugiego, a co za tym idzie również zmianę prędkości, sił, momentów czy też nawet zmianę charakteru ruchu. Przekładnie są nierozłączną częścią maszyn napędzanych silnikiem, czy to spalinowym, czy też elektrycznym. Jest to podyktowane wymaganiami funkcjonalnymi, konstrukcyjnymi oraz gospodarczymi. Niejednokrotnie połączenie bezpośrednie pomiędzy silnikiem, a maszyną wykonującą dla nas określoną pracę jest utrudnione, a nawet niemożliwe, aby zapewnić bezpieczną i wygodną obsługę.

Przekładnie rozróżnia się pod wieloma względami, a mianowicie [6]:

1. Ze względu na liczbę stopni:

* przekładnie jednostopniowe,
* przekładnie wielostopniowe.

1. Ze względu na umiejscowienie zazębienia:

* przekładnie o zazębieniu zewnętrznym,
* przekładnie o zazębieniu wewnętrznym.

1. Ze względu na ruchomość osi:

* przekładnie o osiach stałych,
* przekładnie planetarne.

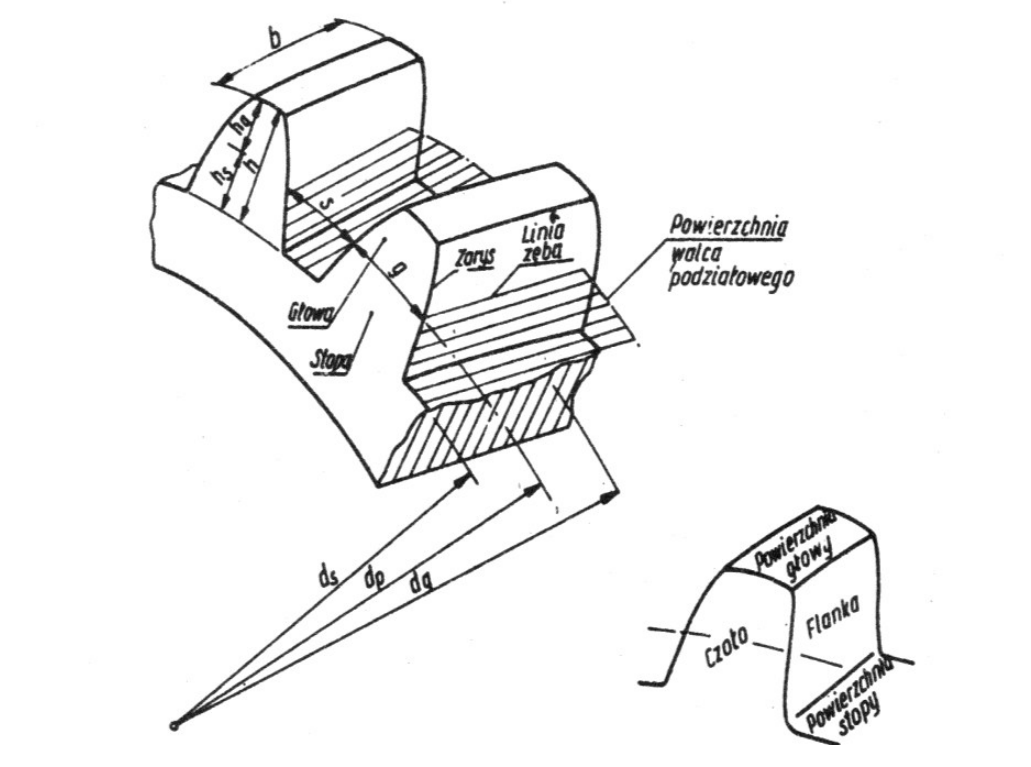
1. Ze względu na kształt kół:

* przekładnie walcowe,
* przekładnie stożkowe,
* przekładnie ślimakowe.

1. Ze względu na wzajemne usytuowanie osi:

* przekładnie równoległe,
* przekładnie kątowe,
* przekładnie wichrowate.

Biorąc pod uwagę mnogość wariantów przekładni zębatych i fakt, że algorytmy obliczeniowe każdego z nich nie są identyczne, postanowiłem zawęzić działanie programu Gear.py i skupić się na jednym rodzaju. Przekładnie zębate, których parametry oblicza program Gear.py to przekładnie jednostopniowe o walcowych kołach zębatych z zazębieniem zewnętrznym, których osie są stałe i ułożone względem siebie równolegle.



Rysunek 8 Geometria i wymiary uzębienia (http://wtie.tu.koszalin.pl/przekladniezebate)

Powyższa grafika obrazuje kształt i wymiary związane z walcowymi kołami zębatymi i ich uzębieniem. Ciężko mówić o obliczeniach przekładni nie wyjaśniając znaczenia charakterystycznych nazw i wymiarów, takich jak:

* ds – średnica stóp
* dg – średnica głów
* dp – średnica podziałowa
* hs – wysokość stopy zęba
* hg – wysokość głowy zęba
* b – szerokość wieńca koła zębatego
* s – szerokość wrębu
* g – grubość zęba
* p – podziałka obwodowa, będąca długością łuku koła podziałowego zawartego między jednoimiennymi sąsiednimi bokami zębów.
* z – liczba zębów
* moduł nominalny – miara wielkości zęba wyrażona w [mm]

Wartości modułu są znormalizowane:

Zalecane wartości w milimetrach: 1; 1,25; 1,5; 2; 2,5; 3; 4; 5; 6; 8; 10; ...

Dopuszczalne wartości w milimetrach: 1,125; 1,375; 1,75; 2,25; 2,75; 3,5; 4,5; 5,5 ...

Zatem:

Założenia początkowe:

* moc przekładni [km]
* prędkość obrotowa [obr./min.]
* przełożenie
* trwałość godzinowa
* maszyna napędzająca
* maszyna napędzana

Zanim przystąpi się do obliczania parametrów przekładni zębatych, warto również pochylić się chwilę nad tym jak rozkładają się siły w zazębieniu w trakcie pracy kół zębatych przekładni. Podczas przenoszenia momentu obrotowego, zęby w kołach przekładni są obciążane zmiennymi naprężeniami wywołanymi siłą normalną Fn, są to naprężenia stykowe σH oraz zginające σF odpowiednio na powierzchni zęba i w jego przekroju. Wspomniane wyżej naprężenia są powodem zmęczeniowego niszczenia zębów przekładni zębatej, w wyniku czego w przekładniach zębatych zamkniętych dochodzi do tzw. „pittingu” lub też do złamania zęba. Warunki wytrzymałościowe, które zabezpieczą projektowaną przez nas przekładnię przed tego typu zniszczeniami to:

* Zabezpieczenie przed „pittingiem”:

σH ≤ σHP

gdzie σHP to dopuszczalne naprężenia stykowe

* Zabezpieczenie przed złamaniem zmęczeniowym zębów:

σF ≤ σFP

gdzie σFP to dopuszczalne naprężenia gnące.

Drugim typem przekładni zębatych są przekładnie otwarte. W tego typu przekładniach zniszczenie zębów zachodzi poprzez zużycie ścierne. Pomimo braku dokładnej metody obliczeń na zużycie ścierne, wiadomo, że zależy ono przede wszystkim od nacisków powierzchniowych. Mając to na uwadze do obliczeń przekładni otwartych jako warunek wytrzymałościowy przyjęto:

σH ≤ σHP

Przed przystąpieniem do obliczeń przekładni zębatych warto zapoznać się z podstawowymi zależnościami kinematycznymi i geometrycznymi, które umieściłem w poniższej tabeli.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parametr** | | | **Jednostka** | **Zależność** |
| moduł normalny | | | mm | mn |
| odległość osi | | | mm | ɑ w = 0,5(dw1(2)+dw2) |
| przełożenie | | | - | u = n1/n2 = dw2/dw1 = z2/z1 |
| kąt pochylenia linii zęba | | o zębach prostych | ° | β = 0° |
| o zębach skośnych | ° | β = (8°…22°) |
| średnice okręgów kół zębatych | tocznych | | mm | dw1(2) = mnz1(2)/cosβ |
| wierzchołków zębów | | da1(2) = mn(z1(2)+2)/cosβ |
| stop zębów | | df1(2) = mn(z1(2)-2,5)/cosβ |
| zasadniczych | | db1(2) = dw1(2)cosα |
| szerokość wieńca | koła zębatego | | mm | B2 = ψbddw1 = ψbɑɑw |
| zębnika | | B1 = b2 + (3…5)mm |
| prędkość kół | | | m/s | ϑ = πdw1n1/(60∙103) |

Tabela 1 Podstawowe zależności geometryczne i kinematyczne walcowych przekładni zębatych

## Dobór materiału

Projektowanie to bardzo złożony proces, w którym na wielu etapach musimy przekładać swoje idee i wymagania rynku na szczegółową wiedzę niezbędną do wykonania końcowego wyrobu. Niewątpliwie, niezwykle ważnym etapem jest dobór odpowiedniego materiału, z którego wykonany zostanie nasz projekt. Wahałem się określając dobór materiału jako etap, ponieważ myślę, że w projektowaniu inżynierskim rzadko zdarza się, aby autor w trakcie kolejnych faz rozwoju projektu nie zmienił wizji co do tego jaki materiał zastosuje do jego realizacji.

Żyjemy w czasach, w których inżynierowie dysponują ogromną ilością materiałów, szacowaną na dziesiątki tysięcy. Standaryzacja dąży to zmniejszania tej liczby, jednak technologia rozwija się tak szybko, że ciągle pojawiają się nowe materiały, które oferują coraz to nowsze, bardziej użyteczne właściwości. Ciężko zatem, z tak szerokiej gamy od razu wybrać konkretny materiał, mając na uwadze na jak wiele czynników będzie to miało wpływ. Od tej decyzji będzie zależeć proces kształtowania, łączenia, wykończenia czy też innego sposobu obróbki końcowego produktu. Można więc stwierdzić, że to jaki materiał wybierzemy, będzie miało wpływ na cały proces technologiczny, a co za tym idzie także koszt naszego projektu i w przyszłości jego konkurencyjność na rynku. Do tego wszystkiego dochodzi jeszcze fakt, że poprawna konstrukcja i wykonanie wyrobu nie gwarantują nam sukcesu sprzedażowego.

Początkowe fazy projektu charakteryzują się „płynnością”, ponieważ nigdy nie jesteśmy w stanie przewidzieć już na początku wszystkich problemów, które pojawią się w kolejnych etapach. Tak samo należy podejść do doboru odpowiedniego materiału – z dozą „płynności”, czyli warto określić sobie pewien zakres materiałów, z których chcielibyśmy korzystać i wraz z zaostrzaniem się kryteriów, gdy projekt nabiera realnych kształtów, lista materiałów również powinna ulec uszczupleniu, aż znajdziemy ten najbardziej nam odpowiadający.[5]

Najczęściej wybieranym materiałem do produkcji kół zębatych są stale poddawane obróbce cieplnej. Stal jest to nic innego jak stop żelaza z węglem. Jeżeli zawartość węgla nie przekracza 2% to mówimy o stali węglowej, natomiast często do takiego stopu dodawane są również umyślnie inne pierwiastki, aby uzyskać pożądane właściwości, wtedy taką stal nazywamy stopową. Można również spotkać koła zębate wykonane z żeliwa, stopów metali kolorowych, a także tworzyw sztucznych.

Ważnym aspektem

# Implementacja algorytmu obliczeniowego

# Pozostałe funkcje

# Weryfikacja poprawności obliczeń na przykładzie

# Podsumowanie, wnioski i uwagi końcowe

# Bibliografia

1. Qt Designer Manual <https://doc.qt.io/qt-5/qtdesigner-manual.html> dostęp: 13.04.2021
2. Qt5 Python GUI Programming
3. Scscs
4. Leonid Kurmaz Projektowanie Węzłów i części maszyn
5. Ashby Dobór Materiałów w Projektowaniu Inżynierskim.
6. Czarnigowski J.: [Wykład 8 przekładnie zębate część 1](http://www.pkm.pollub.pl/index.php?kat=153&page=1&rid=468#id468/). Politechnika Lubelska. Wykłady dla studiów zaocznych. [dostęp 2021-05-16].
7. [http://wtie.tu.koszalin.pl/przekladniezebate dostęp 16.05.2021](http://wtie.tu.koszalin.pl/przekladniezebate%20dostęp%2016.05.2021)

# Spis tabel i rysunków

[Rysunek 1MITCalc Spur Gear - interfejs użytkownika 7](#_Toc69233792)

1. <https://www.gearbox.com.pl/pl/index_pl.html> dostęp: 13.04.2021. [↑](#footnote-ref-1)
2. <https://pypi.org/project/PyQt5/> dostęp: 17.04.2021r. [↑](#footnote-ref-2)