

**Państwowa Wyższa Szkoła Zawodowa  
im. Jana Amosa Komeńskiego  
w Lesznie  
Instytut Politechniczno-Rolniczy**

Kierunek studiów: Elektrotechnika  
Specjalność: Elektromechatronika

Dawid Herkt

**Badanie efektywności pracy regulatora PID  
w układzie regulacji z systemem kulka-belka**

**Analysis of PID regulator in the control system  
with a ball and beam process**

Praca dyplomowa inżynierska

Kierujący pracą:  
dr hab. inż. Sławomir Stępień, prof. nadzw.

Leszno, 2017 r.



**PAŃSTWOWA WYŻSZA SZKOŁA ZAWODOWA  
im. J. A. Komeńskiego w Lesznie**

**Leszno, dn. 23.10.2017 r.**

<b>Studia -</b>	stacjonarne inżynierskie
<b>Instytut -</b>	POLITECHNICZNO- ROLNICZY
<b>Temat pracy dyplomowej -</b>	inżynierskiej
<b>Dyplomant -</b>	<b>Dawid Herkt</b>
<b>Kierunek studiów -</b>	<b>Elektrotechnika</b>
<b>Moduł/ Specjalność -</b>	<b>Elektromechatronika</b>
<b>1. Temat pracy –</b>	Badanie efektywności pracy regulatora PID w układzie regulacji z systemem kulka-belka
<b>2. Dane wyjściowe –</b>	Literatura dotycząca sterowania i układów regulacji
<b>3. Zadania szczegółowe –</b>	Wykonanie projektu oraz budowa układu regulacji kulka-belka. Zaprogramowanie urządzenia. Dobranie nastaw regulatora PID. Przeprowadzenie badań efektywności pracy regulatora PID.
<b>4. Miejsce przeprowadzania analiz lub prac badawczych (laboratoryjnych, konstruktorskich i innych) -</b>	Instytut Politechniczno- Rolniczy
<b>5. Termin oddania pracy -</b>	
<b>6. Kierujący pracą / konsultant / -</b>	dr hab. inż. Sławomir Stępień, prof. nadzw.
<b>7. Recenzent-</b>	

**Kierujący pracą dyplomową**

**Dyrektor Instytutu**

---

dr hab. Sławomir Stępień, prof. nadzw.

## **STRESZCZENIE**

Celem pracy było zbadanie efektywności pracy regulatora PID. Zakres przedmiotowy pracy obejmował projekt oraz budowę układu kulka-belka, na którym zbadana została w sposób eksperymentalny efektywność pracy regulatora PID. W pierwszym rozdziale przedstawiono historię regulatora oraz jego istotę jako elementu układu sterowania automatycznego. Omówiono klasyfikacje systemów regulacji. Scharakteryzowano regulator PID i wybrane metody doboru nastaw regulatora. W drugim rozdziale zaprezentowano model układu kulka-belka oraz realizację projektu pod względem mechanicznym, elektrycznym i programistycznym. W trzecim rozdziale przeprowadzono badanie regulatora. Całość pracy podsumowano w zakończeniu.

## **SUMMARY**

The thesis aims at examine of the PID control efficiency. The scope includes project making, ball on beam device building and the controllers work experimentally testing. The first part includes the PID control system and his role as a part of the automatic control system. In this work classification of control systems is presented and discussed. Moreover the control methods and tuning techniques are described. The second part deals with a device design, considering mechanics, electric circuit and control application. The third part includes a tests for the PID controller working. The whole thesis summitted in overview.

## **Spis treści**

WSTĘP .....	6
CEL I ZAKRES PRACY .....	8
1. POJĘCIE REGULATORA .....	8
1.1. Historia regulatorów .....	9
1.2. Regulator jako element układów sterowania automatycznego.....	9
1.3. Wielkości charakteryzujące regulator .....	9
1.4. Klasyfikacja systemów regulacji .....	10
1.5. Regulator PID .....	10
1.6. Wybrane metody doboru nastaw regulatora .....	11
2. PROJEKT I REALIZACJA URZĄDZENIA SŁUŻĄCEGO DO BADANIA REGULATORA PID .....	12
2.1. Konstruowanie układu kulka-belka w programie SolidWorks.....	12
2.2 Realizacja zaprojektowanego obiektu sterowania.....	18
2.2.1. Ujęcie mechaniczne .....	18
2.2.2 Ujęcie elektryczne.....	23
2.2.3 Ujęcie programistyczne .....	34
3. BADANIE REGULATORA.....	41
3.1. Strojenie ręczne .....	41
3.2 Wyznaczanie nastaw metodą Zieglera-Nicholsa .....	43
ZAKOŃCZENIE.....	47
BIBLIOGRAFIA.....	49
SPIS TABEL .....	50
SPIS RYSUNKÓW .....	50
SPIS ZAŁĄCZNIKÓW .....	51

## **WSTĘP**

Wybór tematu pracy dyplomowej inżynierskiej „Badanie efektywności pracy regulatora PID w układzie regulacji z systemem kulka-belka” motywowany jest zainteresowaniem autora tematyką związaną z automatyką, elektroniką oraz programowaniem mikrokontrolerów.

W przemyśle powszechnie stosowane są urządzenia i systemy automatyczne. Pomagają one zoptymalizować procesy produkcyjne skracając czas oraz obniżając koszty produkcji. Systemy sterowania automatycznego bardzo wcześniej rozwijane są na potrzeby militariów. Wykorzystanie przy produkcji broni najnowocześniejszych technologii może być decyzyjne dla wyniku konfliktów (np. systemy rakietowe wykrywające, identyfikujące, śledzące i zwalczające cele powietrzne). Z automatyką spotykamy się jednak nie tylko w przemyśle i wojsku. Coraz więcej urządzeń i systemów automatycznych wykorzystywanych jest przez ludzi w codziennym życiu, poprawiając m.in. komfort życia (np. automatyczna klimatyzacja), czy bezpieczeństwo (np. system ABS stosowany w samochodach). Dynamiczny rozwój komputerów oferujących coraz większą moc obliczeniową pozwala na budowę coraz bardziej zaawansowanych systemów automatycznych, które zastępują ludzi przy wykonywaniu coraz bardziej skomplikowanych czynności (np. autonomiczne pojazdy).

Układy sterowania automatycznego można podzielić na dwa rodzaje:

- układy otwarte sterowania automatycznego – to grupa układów, w których nie występuje pętla sprzężenia zwrotnego. Z uwagi na to wymagane są kompletne informacje o sterowanym obiekcie. Takie układy realizują zadaną kolejność działań niezależnie od zakłóceń działających na obiekt, czy też stanu wielkości sterowanej. Otwarte układy sterowania automatycznego powszechnie wykorzystywane są w automatach cyklicznych oraz obrabiarkach numerycznych. Maszyny te kolejno wykonują zadania zapisane przez operatora na nośnikach pamięci.
- układy zamknięte sterowania, czyli układy regulacji automatycznej – układy te działają w oparciu o zasadę ujemnego sprzężenia zwrotnego. Wartość zadana porównywana jest z wartością wielkości regułowanej. Wynikiem porównania jest sygnał uchybu uwzględniany w procesie sterowania. W przeciwieństwie do układów otwartych, kompletna znajomość informacji początkowych o regułowanym procesie nie jest konieczna.

Z kolei układy regulacji automatycznej dzielą się na trzy grupy w zależności od sposobu wprowadzania wartości zadanych wielkości regułowanych. Wyróżnić można układy regulacji:

- stałowartościowej (układy stabilizujące),
- programowej,
- nadążnej (układy śledzące) [1].

Najczęściej spotykane są układy stabilizujące. W tych układach wartość zadana jest stała, może być zmieniona jedynie z zewnątrz przez operatora. Z kolei w układach regulacji programowej wartość zadania zmienia się w czasie według programu wprowadzonego przez obsługę. Natomiast układy śledzące nie posiadają zadajnika. Układ w czasie rzeczywistym określa wartość zadaną na podstawie obserwacji zewnętrznych zjawisk. W praktyce często stosuje się układy kombinowane, które zawierają elementy z różnych rodzajów układów sterowania. Przykładem takiego układu może być tempomat adaptacyjny. Układ ten stara się utrzymać stałą, zadaną przez kierowcę prędkość niezależnie od nachylenia nawierzchni, czy prędkości i kierunku wiatru. W takiej sytuacji działa jak układ regulacji stałowartościowej, jednak tempomat adaptacyjny mierzy dystans do pojazdów poruszających się przed pojazdem, w którym jest zainstalowany. Po przekroczeniu ustalonej odległości między pojazdami, tempomat działa jak układ regulacji nadążnej. Wartość zadana jest automatycznie zmieniana, prędkość zostaje dostosowana, aby zapewnić bezpieczną odległość między pojazdami [2].

Zasadniczym elementem każdego układu regulacji automatycznej jest regulator. Jego zadaniem jest generowanie sygnału sterującego, który oddziałuje na obiekt sterowany w pożądany sposób [3].

Najczęściej stosowanym regulatorem jest regulator proporcjonalno-całkującoco-różniczkujący PID. Jest on implementowany w różnych formach. Może być częścią większego systemu sterowania, może funkcjonować również samodzielnie [4].

Regulator PID pracuje w układzie ze sprzężeniem zwrotnym ujemnym. Oznacza to, że sygnał wyjściowy regulatora oddziałuje na sygnał wejściowy. Sprzężenie zwrotne ujemne ma charakter stabilizujący. Skutki działania regulatora przeciwdziałają jego przyczynie [5].

Najprostszym regulatorem ze sprzężeniem zwrotnym jest regulator dwustanowy włącz-wyłącz. Sygnał wyjściowy takiego regulatora zawsze ma wartość maksymalną lub minimalną, co często powoduje oscylacje wokół wartości zadanej. Powodem są zbyt duże wartości sterujące w odpowiedzi na niewielki uchyb. Takiej sytuacji przeciwdziała regulator proporcjonalny, którego odpowiedź jest proporcjonalna do uchybu. Efektywność pracy regulatora proporcjonalnego można zwiększyć wzmacniając sygnał sterujący, jednak zbyt duże wzmacnienie sygnału prowadzi do przeregulowania. Regulator proporcjonalny jest składową regulatora PID.

Natomiast głównym zadaniem części całkującej regulatora PID, jest osiągnięcie wartości zadanej w stanie ustalonym. Efektem pracy regulatora proporcjonalnego jest ustalenie wartości sterowanej bliżej wartości zadanej, jednak zawsze występuje niewielki błąd. Część całkująca sumuje małe błędy, zwiększając odpowiedź regulatora na niewielkie uchyby, dzięki czemu układ stabilizuje się dokładnie w wartości zadanej.

Zadaniem części różniczkującej jest zwiększenie stabilności pracy układu. Od momentu zarejestrowania wartości uchybu do wykonania akcji przez mechanizm mija czas, w którym wartość uchybu już się zmieniła. Dlatego układ zawsze jest spóźniony, jego odpowiedź jest nieodpowiednia do aktualnego stanu. Takiej sytuacji przeciwdziała część różniczkująca. Dodaje ona wartość do sygnału sterującego, zależną od tego jak szybko wartość uchybu zmienia się w jednostce czasu. Dzięki temu układ sterowany jest na przewidywaną wartość uchybu, w momencie wykonania odpowiedzi [6].

## **CEL I ZAKRES PRACY**

Celem pracy było zbadanie efektywności pracy regulatora PID w układzie regulacji z systemem kulka-belka.

Zakres pracy obejmował część teoretyczną, zawartą w pierwszym rozdziale niniejszej pracy; część praktyczną, opisaną w rozdziale drugim oraz analityczną opisaną w trzecim rozdziale. Część teoretyczna zawiera podstawowe informacje na temat systemów regulacji automatycznej oraz samego regulatora PID. Omówiono również wybrane metody doboru nastaw regulatora. Część praktyczna obejmuje projekt układu kulka-belka, dobór elementów składających się na urządzenie, budowę układu i zaimplementowanie na mikrokontrolerze regulatora PID oraz pozostałych funkcji obsługujących wszystkie podzespoły urządzenia. Natomiast w części analitycznej przedstawiono doświadczalne wyznaczenie nastaw regulatora PID oraz analizę wyników przeprowadzonych badań.

Cele szczegółowe pracy obejmują:

- zaprojektowanie układu kulka belka,
- dobór elementów składających się na układ,
- zbudowanie urządzenia,
- napisanie programu sterującego układem z graficznym interfejsem wyświetlającym wykres w czasie rzeczywistym,
- dobór nastaw regulatora sposobami doświadczalnymi,
- zbadanie efektywności pracy regulatora PID dla dobranych nastaw.

W pracy dyplomowej zostały wykorzystane źródła wtórne, tj. literatura i artykuły naukowe oraz pierwotne, uzyskane przez autora w wyniku wykonanych na układzie regulacji z systemem kulka-belka pomiarów.

Praca ma charakter projektowy i badawczy. Zawiera trzy rozdziały poprzedzone wstępem oraz celem i zakresem pracy, i podsumowane zakończeniem.

W pierwszym rozdziale skupiono się na pojęciu regulatora. Przybliżono historię regulatorów. Opisano pierwsze zbudowane regulatory. Następnie przedstawiono regulator jako składową układów sterowania automatycznego. Opisano zasadnicze elementy wchodzące w skład układu regulacji automatycznej oraz pełniące przez nie funkcje. Omówiono również wielkości charakteryzujące regulator. Wyjaśniono różnicę między wielkościami sterowanymi, a sterującymi. Wyodrębniono i scharakteryzowano podział systemów regulacji pod względem zadań, rodzaju elementów, z których składa się układ, sposobu pomiaru wielkości regułowanej oraz sposobu przekazywania sygnałów. Przedstawiono regulator PID. Scharakteryzowano jego strukturę. Omówiono wybrane metody doboru nastaw regulatora.

W drugim rozdziale przedstawiono proces projektowania w programie SolidWorks. Omówiono kroki, które należy wykonać w trakcie tworzenia modelu trójwymiarowego. Opisano budowę układu pod względem mechanicznym, elektrycznym i programistycznym. Zaprezentowano części wchodzące w skład układu oraz metodykę doboru tych elementów. Zobrazowano również zadania jakie wykonują moduły zastosowane w urządzeniu. Przybliżono zastosowane metody wyznaczania pozycji kulki na belce oraz kąt nachylenia belki. Scharakteryzowano język programowania Python, w którym został napisany program. Opisano biblioteki użyte w programie oraz wielowątkowy algorytm sterowania układem. Zobrazowano zaimplementowane funkcje obsługujące podłączone podzespoły, obliczające wartości regulatora PID oraz wyświetlające interfejs graficzny. Przedstawiono interfejs graficzny, zbudowany za pomocą modułu Tkinter oraz sposób wyświetlania wykresu w czasie rzeczywistym.

Natomiast w trzecim rozdziale przedstawiono proces doboru nastaw regulatora dwoma metodami doświadczalnymi. Następnie porównano wyniki badania efektywności pracy układu przy nastawach uzyskanych za pomocą strojenia ręcznego z nastawami uzyskanymi metodą Zieglera-Nicholsa. Wyniki zestawiono na wykresach. Dokonano analizy uzyskanych rezultatów, na podstawie których opracowano i przedstawiono wnioski.

# **1. POJĘCIE REGULATORA**

## **1.1. Historia regulatorów**

Początków automatyki przemysłowej można doszukiwać się w 1784 roku. Wtedy brytyjski inżynier James Watt otrzymał patent na odśrodkowy regulator do stabilizacji prędkości obrotowej maszyn parowych. Następnym odnotowanym użyciem regulatora było automatyczne regulowanie ciśnienia gazu w sieci oświetlenia ulic Londynu. W 1830 roku wynaleziono bimetaliczny regulator temperatury. Natomiast około dwadzieścia trzy lata później Amerykanin Sickel opatentował hydrauliczny serwomechanizm sterujący maszyną parową na statku.

W 1907 roku nowojorska firma C.J. Tagliabue Company wykorzystała pneumatyczny dwustawny regulator temperatury sterujący procesem pasteryzacji mleka. W tym samym roku G. Dalen skonstruował układ automatycznej regulacji wielkości płomienia i częstotliwości jego pulsacji, za co pięć lat później przyznano mu nagrodę Nobla. Wynalazek ten stosowany był w bojach świetlnych oraz latarniach morskich. W pierwszej połowie XX wieku stosowano regulatory pneumatyczne, w których rolę czujnika pełniła rurka Bourdona. Deformacja rurki miała wpływ na stopień otwarcia zaworu. System ten nie charakteryzował się dokładnością, co było motywatem dla Edgara H. Bristol do zaproponowania w 1914 r. stosowania wzmacniacza dysza-przesłona.

W 1922 roku prace N. Minorskiego, dotyczące stabilności układów nieliniowych, zlecone przez amerykańską marynarkę wojenną. Finalizacją tych prac było zastosowanie regulatora PID (proportional–integral–derivative controller, czyli regulator proporcjonalno-całkującoczo różniczający) do sterowania kursem statku [9].

## **1.2. Regulator jako element układów sterowania automatycznego**

Do zespołu układów sterowania automatycznego należą: obiekt sterowania, regulator, element nastawczy oraz miernik. Obiektem sterowania, inaczej regulacji, można nazwać proces, urządzenie lub układ dynamiczny, charakteryzujący się tym, że można dzięki niemu wymusić pożądane stany. Dzieje się tak w wyniku oddziaływanego obiektu sterowania pomimo występujących w otoczeniu zakłóceń. Natomiast urządzenie sterujące (regulator) porównuje „wartość sygnału zadanego z wartością sygnału sterowanego (regulowanego) obiektu oraz na podstawie ich różnicy wytwarzający odpowiedni sygnał sterujący obiektem”. Do zasadniczych elementów regulatora należą:

- zadajnik, czyli funkcja lub urządzenie, za pomocą której do regulatora podawana jest wartość zadana
- sumator, którego funkcją jest porównanie wartości sygnału zadanego z wartością sygnału regulowanego. Wynikiem tego działania jest sygnał uchybu regulacji
- wzmacniacz uchybu, którego zadaniem jest wzmacnianie sygnału uchybu do stopnia umożliwiającego oddziaływanie na element wykonawczy układu
- element wykonawczy układu jest to część funkcjonalna powodująca zmianę położenia elementu nastawczego.

Trzeci z omawianych układów sterowania automatycznego, element nastawczy, jest to urządzenie bezpośrednio oddziaływanie na sterowany proces. Przykładowo może to być siłownik lub silnik. Natomiast miernik to urządzenie, w którego skład wchodzi najczęściej czujnik i przetwornik. Dostarcza on sygnał porównywany w sumatorze z sygnałem zadanym [7].

## **1.3. Wielkości charakteryzujące regulator**

Wielkości sterowane to wielkości wobec których określa się cel sterowania. Wielkości te zazwyczaj można mierzyć lub uzyskiwać poprzez wykorzystanie innych pozyskiwanych informacji. Natomiast wielkości sterujące to wielkości, które są możliwe do modyfikacji

z zewnętrz. Oddziałują one na wielkości sterowane. Decyzje sterujące stanowią ich wartości. Dzięki prawidłowej selekcji i modyfikacji wielkości sterujących wykonuje się sterowanie. Zakłócenia charakteryzują wpływ otoczenia na wielkości sterowane i sterujące. Sterowanie opiera się zazwyczaj na reakcji przeciwko zakłóceniom. Zakłócenia te mogą być mierzalne lub niemierzalne. Te pierwsze mogą być mierzone w celu ich wykorzystania podczas podejmowania decyzji sterujących [1].

#### **1.4. Klasyfikacja systemów regulacji**

Systemy regulacji można sklasyfikować między innymi ze względu na zadania, rodzaj elementów, z których składa się układ, sposób pomiaru wielkości regulowanej, sposób przekazywania sygnałów.

Pierwszy podział, pod względem realizowanych zadań wyróżnia układy adaptacyjne, nadążne, programowe oraz stabilizacji. Układy adaptacyjne, inaczej nazywane autostrojącymi, ekstremalnymi lub autoptymalizującymi polegają na sterowaniu obiektem, który posiada zmienne zakłócenia i właściwości dynamiczne. W trakcie ich oddziaływanego szacuje się wartości modelu obiektu oraz zakłóceń, aktualizując w ten sposób parametry algorytmu sterowania. Te drugie układy, nadążne charakteryzują się tym, że celem układu jest nadążanie wartości wyjściowej za zmieniającą się w sposób nieznany wielkością zadaną. Natomiast w przypadku układów programowych wartość zadana zmienia się w czasie według określonego programu. Tego rodzaju układy wykorzystywane są zazwyczaj w obrabiarkach sterowanych numerycznie. Ostatni rodzaj układów pod względem realizowanych zadań to układy stabilizujące. Cel układu to utrzymanie niezmiennej wartości wielkości wyjściowej i zmniejszenie wpływu zakłóceń na te wielkości.

Natomiast klasyfikacja pod względem rodzaju elementów, z których składa się układ wyróżnia układy liniowe i nieliniowe. Różnią się one tym, że te pierwsze zawierają wyłącznie elementy liniowe, w których równania różniczkowe mają stałe współczynniki. Natomiast nieliniowe zawierają przynajmniej jeden element nieliniowy. Za układ nieliniowy można uznać przykładowo przekaźniki i regulatory dwu- i trójpołożeniowe.

Trzeci podział systemów sterowania uwzględnia sposób pomiaru wielkości regulowanej. Wyróżnia się tutaj układy analogowe i cyfrowe. Te pierwsze charakteryzują się tym, iż pomiar następuje tutaj w sposób ciągły, a uzyskane wartości prezentowane są za pomocą funkcji zależności wyniku pomiaru od wielkości regulowanej, np. termometr, manometr. A cyfrowe układy cechuje to, że pomiar przekształcany jest za pomocą przetworników analogowo-cyfrowych.

Ostatni podział systemów regulacji to klasyfikacja według sposobu przekazywania sygnałów. Wyróżnia się tutaj układy regulacji ciągłej i dyskretniej (przerywanej). Te pierwsze charakteryzują się tym, że każda część układu pracuje w sposób ciągły i może przyjmować wszystkie wielkości z zakresu zmienności sygnału.. Natomiast układy regulacji przerywanej zakładają, że przynajmniej jedna część działa w sposób dyskretny. W przypadku wysłania sygnałów przez tą część przyjmują one jedynie określone wielkości, występujące w wybranych chwilach czasu. Takie układy możliwe są do opisania za pomocą równań różnicowych [8].

#### **1.5. Regulator PID**

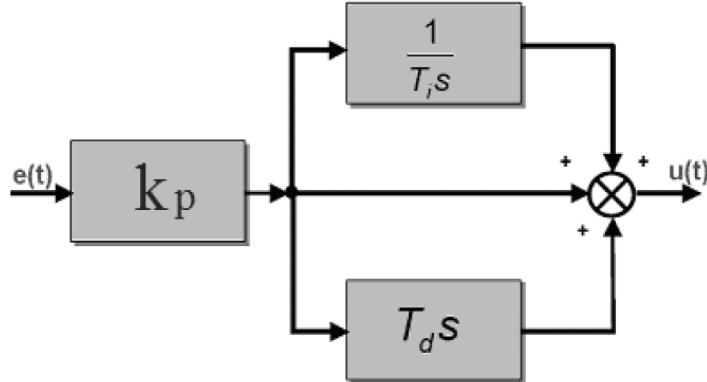
Regulator PID składa się z trzech członów:

- proporcjonalny (proportional)
- całkujący (integral)
- różniczkujący (derivative)

Transmitancja operatorowa idealnego regulatora PID to:

$$G_{PID}(s) = \frac{U(s)}{E(s)} = K_P \left( 1 + \frac{1}{T_I s} + T_D s \right) \quad (1.1)$$

gdzie  $U(s)$  jest odpowiedzią układu na uchyb  $E(s)$ ,  $K_P$  jest wzmocnieniem proporcjonalnym, natomiast  $T_I$  oraz  $T_D$  to stałe czasowe całkowania i różniczkowania, zwane czasem zdwojenia i wyprzedzenia. Ogólną strukturę regulatora PID przedstawiono na schemacie zamieszczonym na rysunku 1 [9].



Rys. 1. Ogólna struktura regulatora PID

Odpowiedź regulatora PID na zakłócenia może być przedstawiona w formie:

$$U(s) = K_P E(s) + K_I \frac{1}{s} E(s) + K_D s E(s) \quad (1.2)$$

gdzie:  $K_I = K_P / T_I$  jest to wzmocnienie całkujące, a  $K_D = K_P T_D$  jest wzmocnieniem różniczkującym. Każdy z elementów regulatora ma inne zadanie:

- proporcjonalny – równoważy bieżący uchyb,
- całkujący – kompensuje uchyb na podstawie uchybów występujących w przeszłości,
- różniczkujący – przewiduje i równoważy uchyby, mające wystąpić w przyszłości [10].

## 1.6. Wybrane metody doboru nastaw regulatora

Istnieje wiele metod doboru nastaw regulatora. Niektóre metody wymagają znajomości modelu matematycznego obiektu, inne opierają się na wykonaniu doświadczeń. Metody doświadczalne wykorzystywane są częściej ze względu na brak konieczności znajomości dynamiki układu oraz większą uniwersalność. Takimi metodami są: strojenie ręczne oraz II Metoda Zieglera-Nicholsa.

Strojenie ręczne polega na ustaleniu regulatora na działanie wyłącznie proporcjonalne. Zwiększeniu wzmocnienia  $K_p$  do momentu uzyskania niegasnących oscylacji. Takie wzmocnienie nazywamy granicznym, lub krytycznym, ponieważ układ znajduje się na granicy stabilności. Kolejnym krokiem jest ustalenie  $K_p$  na połowę wzmocnienia granicznego i zwiększaniu wzmocnienia całkującego  $K_I$  do momentu uzyskania oczekiwanych uchybu. Ostatnim krokiem jest zwiększenie wzmocnienia różniczkującego  $K_D$ , do momentu, gdy czas regulacji będzie wystarczający.

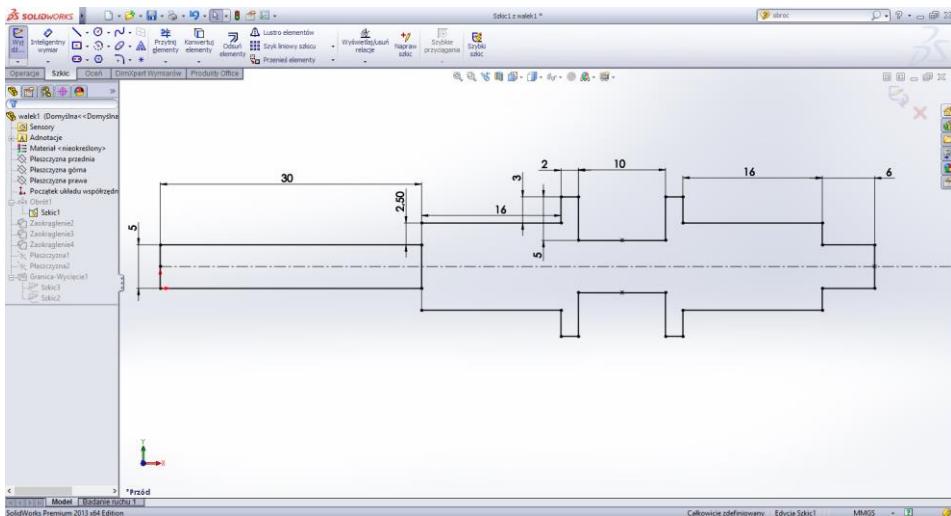
Najczęściej stosowaną metodą doboru nastaw regulatora jest II Metoda Zieglera-Nicholsa. Podobnie jak przy strojeniu ręcznym ta metoda również polega na znalezieniu wzmocnienia granicznego. W tym celu również należy ustawić parametry  $K_I$  oraz  $K_D$  na zero. Zwiększać  $K_p$  do uzyskania niegasących oscylacji. Po uzyskaniu niegasących oscylacji, należy zmierzyć okres drgań, zwany okresem drgań krytycznych, lub granicznych. Wzmocnienie graniczne oznaczone jako  $K_U$  oraz okres drgań granicznych oznaczany jako  $T_U$  służą do wyznaczenia nastaw regulatora według zależności:

$$K_p = 0,6K_U \quad K_I = \frac{1,2K_U}{T_U} \quad K_D = \frac{0,6K_UT_U}{8} \quad (1.3)$$

## 2. PROJEKT I REALIZACJA URZĄDZENIA SŁUŻĄCEGO DO BADANIA REGULATORA PID

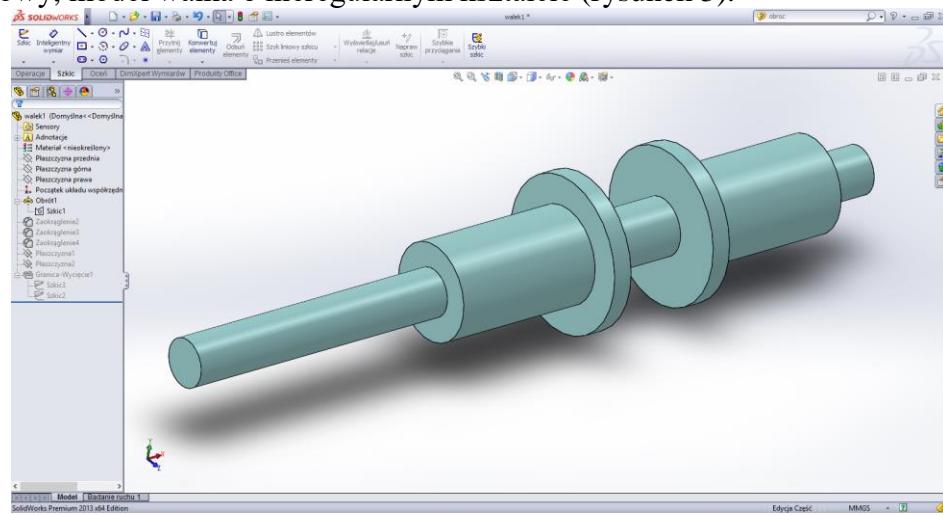
### 2.1. Konstruowanie układu kulka-belka w programie SolidWorks

Układ zaprojektowano w programie SolidWorks Premium 2013. SolidWorks to pakiet kompleksowych rozwiązań do projektowania 3D oraz symulowania ruchu. Daje możliwość testowania zachowania urządzenia będącego w ruchu, pod wpływem działania sił [11]. Konstruowanie w tym programie przebiega w następujący sposób: najpierw rysowany jest szkic, który poprzez wykonywanie operacji „wyciągnięcia szkicu”, „wyciągnięcia przez obrót”, „wyciągnięcia po ścieżce” itd. pozwala na uzyskanie prymitywów. Prymitywy są to podstawowe bryły geometryczne, takie jak prostopadłości, walec, stożek itp., które opisane są parametrami. Następnie bryły te mogą być, na zasadzie algebry Boole'a, dodawane do siebie, odejmowane, a także można uzyskiwać część wspólną kilku prymitywów. Za pomocą tych kilku operacji możliwe jest utworzenie obiektu o dowolnym kształcie. Na rysunku 2. pokazano szkic, który posłużył do wyciągnięcia pierwszego prymitywu tworzącego jedną z części układu.



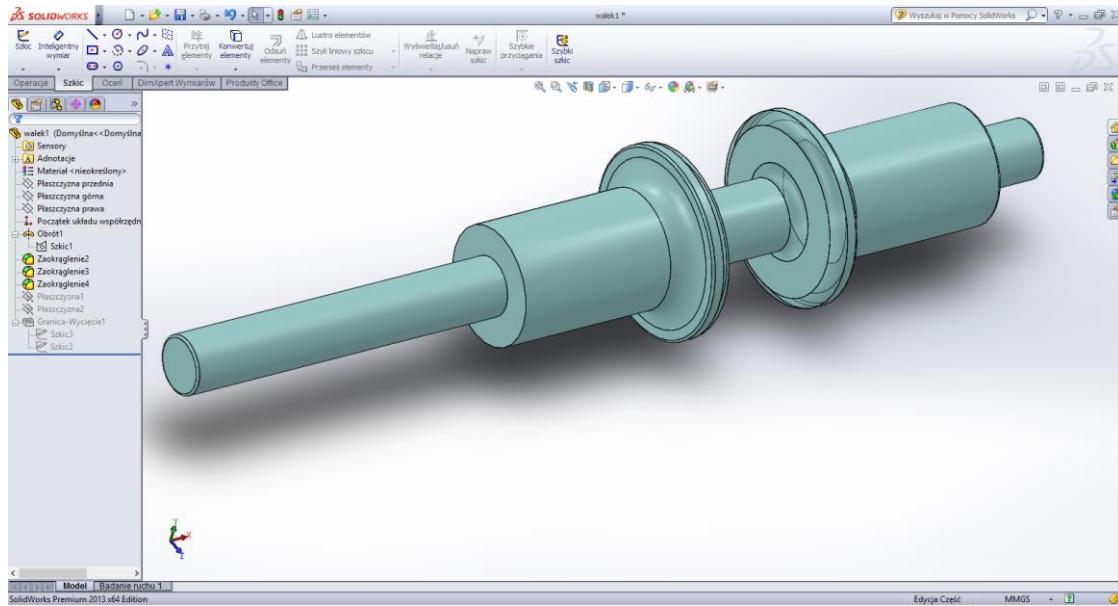
Rys. 2. Szkic części tworzonej w programie SolidWorks Premium 2013

Następnie za pomocą operacji wyciągnięcia przez obrót wokół osi symetrii, uzyskano obiekt trójwymiarowy, model wałka o nieregularnym kształcie (rysunek 3).



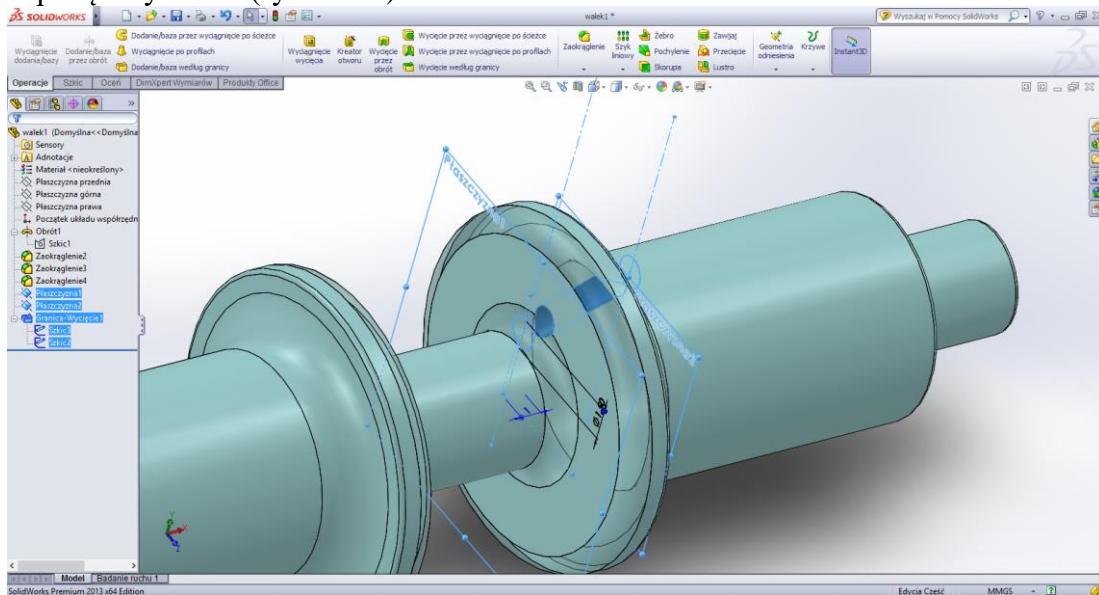
Rys. 3. Model wałka 3D

Kolejnym etapem przy modelowaniu tej części było dodanie zaokrągleń wybranych krawędzi. Efekt tych działań widoczny jest na rysunku nr 4.



Rys. 4. Model wałka 3D z zaokrąglonymi krawędziami

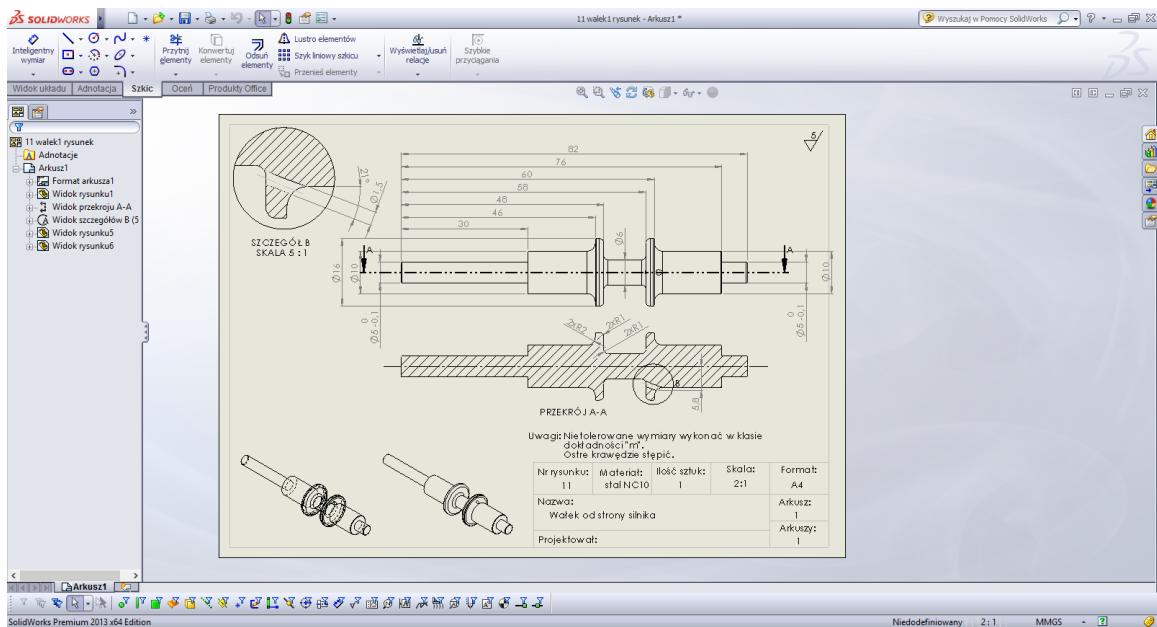
Ostatnim elementem przy projektowaniu tej części było dodanie otworu. W tym celu konieczne było stworzenie dwóch wirtualnych płaszczyzn. Na tych płaszczyznach narysowano okręgi, które zostały wykorzystane przy operacji „wycięcia według granicy”, tym samym tworząc w bryle pożądany otwór (rysunek 5).



Rys. 5. Wycięcie otworu w bryle

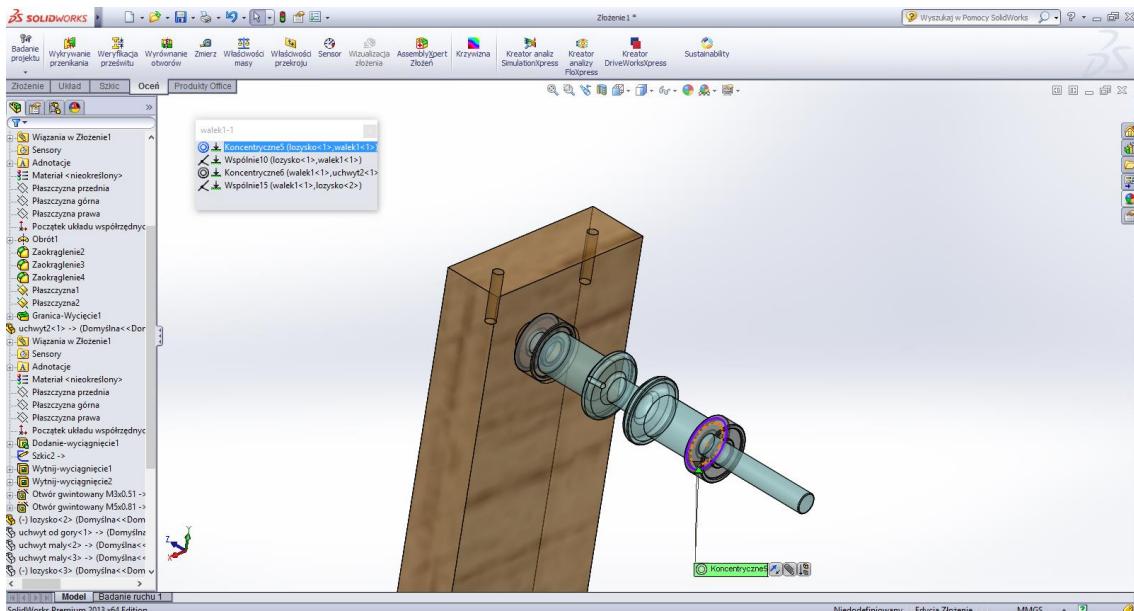
Wszystkie elementy zostały zaprojektowane w podobny sposób. Niektóre z nich, takie jak: silnik krokowy, sprzęgło kłowe, narysowano w uproszczeniu, ponieważ istotne dla projektu były jedynie wymiary zewnętrzne, a nie budowa wewnętrzna elementów. Tak stworzony model części można bardzo szybko przekonwertować na techniczny rysunek wykonawczy. W tym celu konstruktor wybiera odpowiednie widoki i/lub przekroje oraz odpowiednio je rozmieszcz, zgodnie z zasadami rysunku technicznego. Rysunki widoków i przekrojów generowane są automatycznie, jednak wymiarowanie należy dodać ręcznie (rysunek 6). Na podstawie

przygotowanego w taki sposób rysunku technicznego, dana część może zostać wyprodukowana. Rysunki wykonawcze części zostały zawarte w załączniku nr 1.



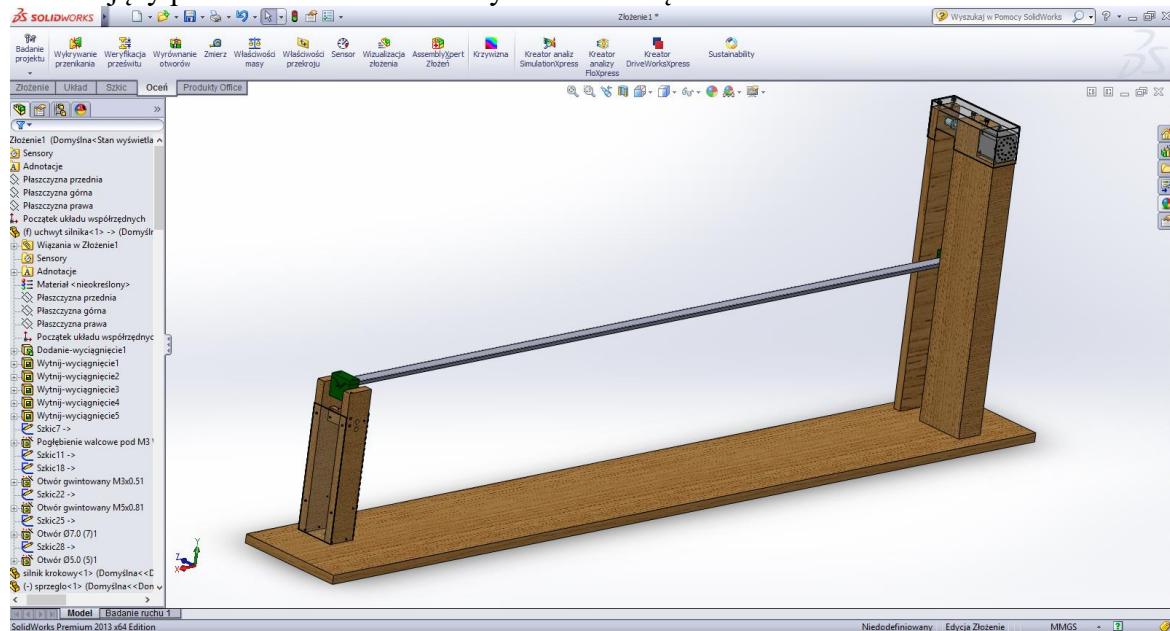
Rys. 6. Gotowy rysunek techniczny wygenerowany w programie

Modele trójwymiarowe części mogą zostać importowane do rysunku złożeniowego. Tworzenie złożenia w programie Solid Works polega na dodawaniu relacji pomiędzy poszczególnymi częściami, np. położenie wałka koncentrycznie do otworu w innej części. Za pomocą tworzenia wiązań między częściami, definiuje się ich względne położenie. W celu zapobieżenia przesuwania się części, warunkiem koniecznym jest jej pełne zdefiniowanie, czyli związanie z innymi częściami złożenia w taki sposób, aby nie mogła zostać przesunięta w żadnej płaszczyźnie (rysunek 7).



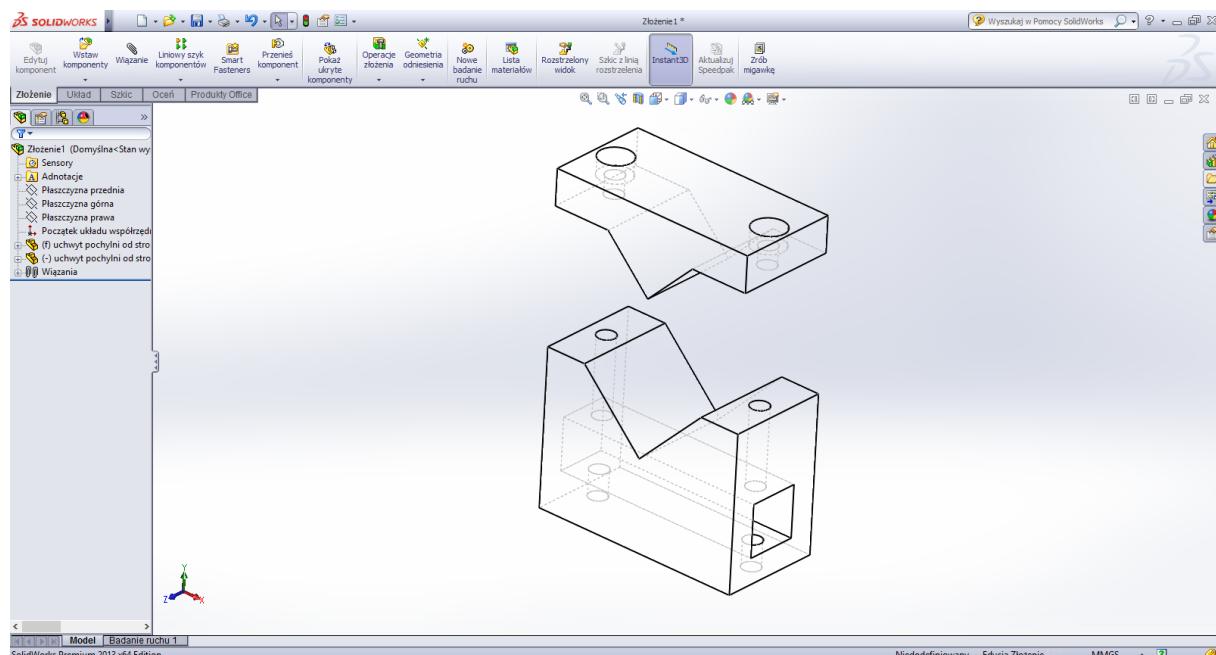
Rys. 7. Wiązanie wałka z łożyskami, oraz otworem pod łożysko

Podczas tworzenia złożenia, nadal istnieje możliwość edytowania części. Ułatwia to dopasowanie elementów względem siebie. Po zapisaniu zmienionego modelu części, zmienia się on automatycznie zarówno w złożeniu, jak i w rysunku technicznym. Dzięki temu wszystkie powiązane pliki są zawsze aktualne. W trakcie budowania układu zmieniana była koncepcja pomiaru pozycji kulki na belce. Z tego względu model układu był kilkukrotnie edytowany. Ostateczny model zaprezentowany został na rysunku nr 8. Plik programu SolidWorks zawierający pełen model mechaniczny zawiera załącznik nr 2.



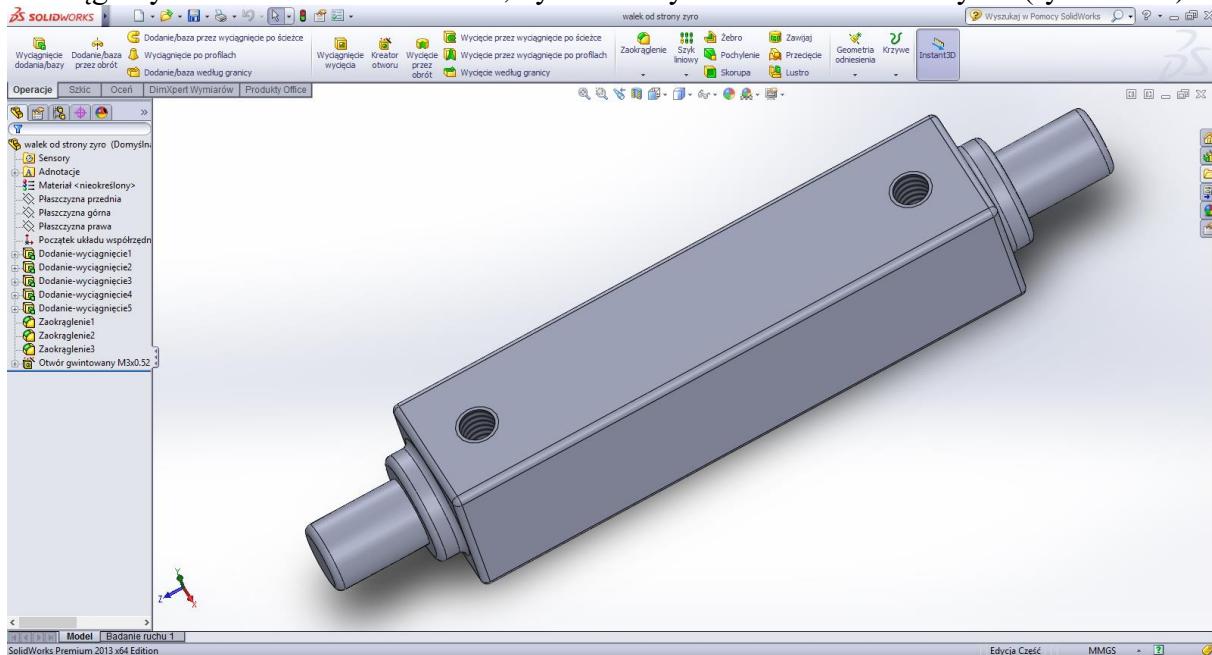
Rys. 8. Pełen model mechaniczny układu kulka-belka

Układ skonstruowano w taki sposób, by zapewnić płynną zmianę kąta pochylni. W tym celu lewy koniec belki został zakotwiczony, umożliwiając poruszanie się belki zaledwie w jednej płaszczyźnie, na zasadzie zawiasu. Zamierzając uzyskać taki efekt, zaprojektowano specjalny uchwyt belki składający się z dwóch części. Poprzez wkręcanie śrub w przygotowane otwory, uchwyt będzie zaciskał się, stabilnie blokując belkę (rysunek 9).



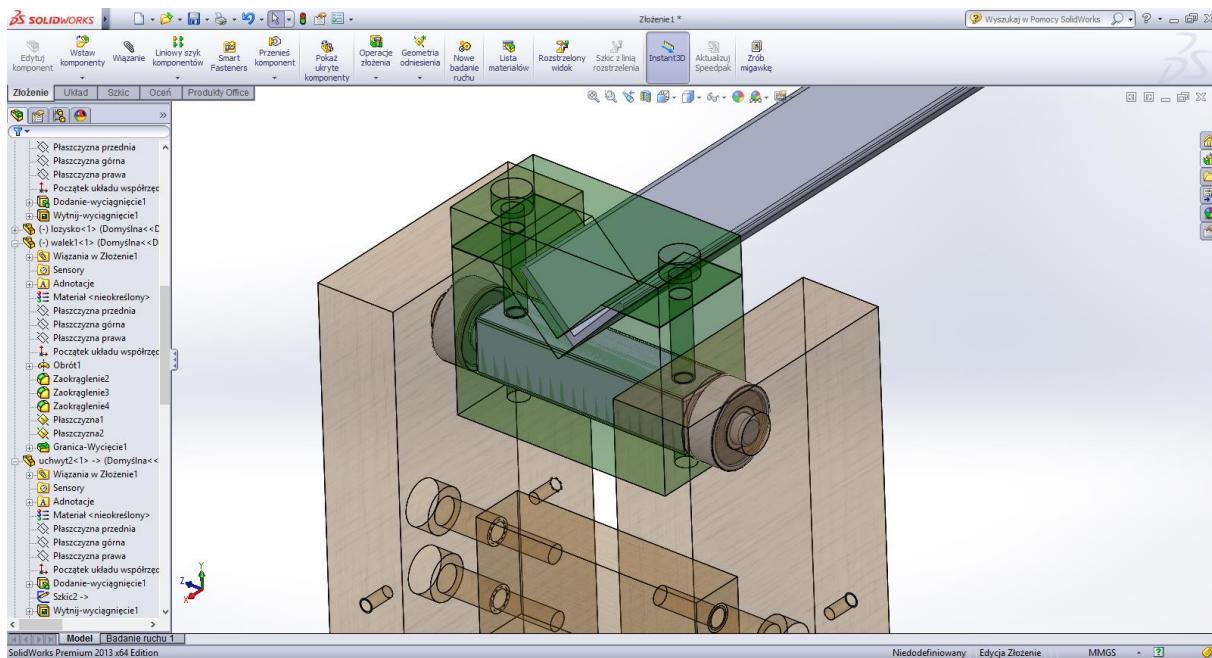
Rys. 9. Projekt uchwytu belki przy niższym ramieniu

Natomiast w kwadratowym otworze zostanie umieszczony stalowy, kwadratowy pręt, zaokrąglony na tokarce z obu stron tak, by można było osadzić na nim łożyska (rysunek 10).



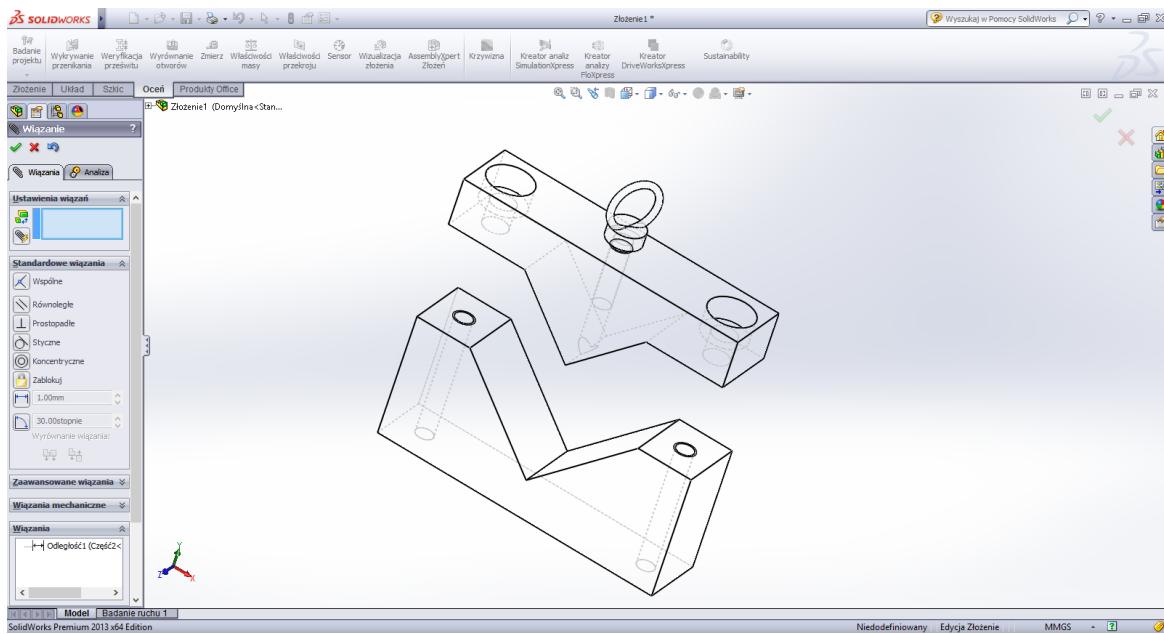
Rys. 10. Projekt specjalnego pręta kwadratowego

Następnie należało przygotowane elementy połączyć. Na rysunku nr 11 zaprezentowano złożenie uchwytu z belką oraz prętem, osadzone na łożyskach w przygotowanych otworach.



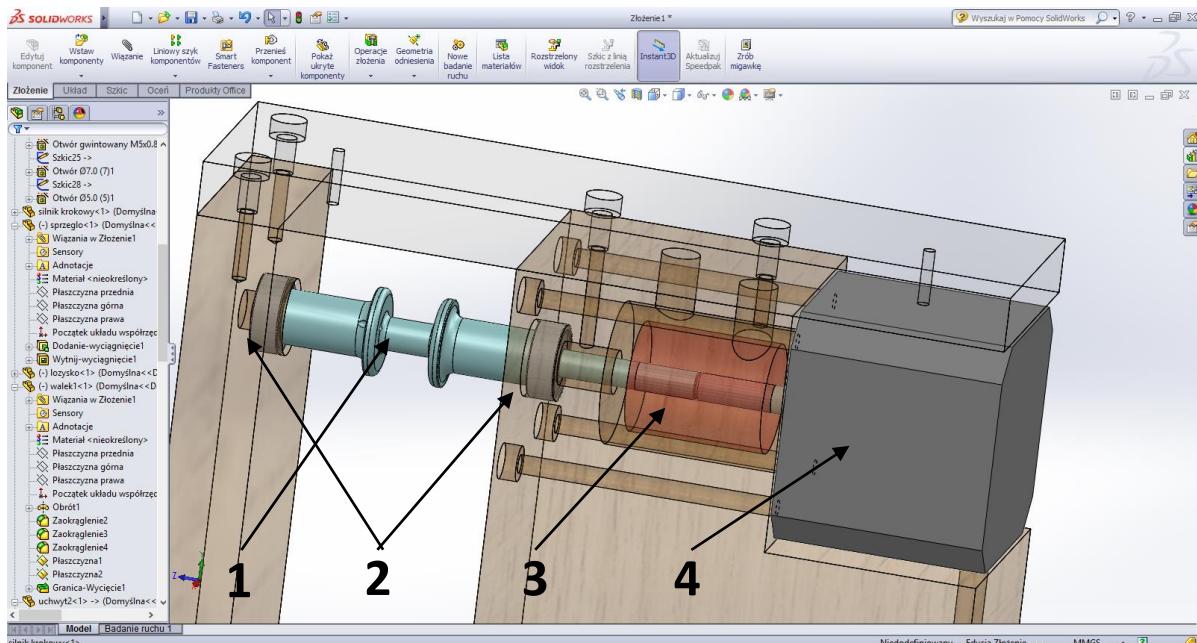
Rys. 11. Złożenie uchwytu z prętem osadzonym na łożyskach w otworach

Natomiast na drugim końcu pochylni również zamontowany zostanie specjalnie zaprojektowany uchwyt, który działać będzie na podobnej zasadzie. Jedynie z tą różnicą, że zamiast kwadratowego otworu pod pręt, znajdująca się będzie w górnej części uchwytu otwór gwintowany, umożliwiający wkręcenie śruby oczkowej. Przez oczko zostanie przewleczona linka. Projekt uchwytu ze śrubą oczkową pokazano na rysunku nr 12.



Rys. 12. Projekt uchwytu belki przy wyższym ramieniu

Za pomocą tak skonstruowanego uchwytu można podwiesić koniec pochylni na lince. Następnie linka zostanie nawinięta na specjalnie wykonany wcześniej wałek. Wałek ten zostanie osadzony na łożyskach w przygotowanych do tego otworach i połączony z silnikiem za pomocą sprzągła kłowego z wkładką elastyczną. Zastosowanie łożysk z obu stron wałka rozłoży obciążenie działające w dół na ramionach konstrukcji, dzięki czemu silnik będzie obciążony jedynie momentem obrotowym. Zastosowanie sprzągła kłowego z elastyczną wkładką pomoże zredukować drgania generowane przez silnik krokowy. Całość będzie działała na zasadzie wciągarki linowej. Szczegółowy model tego fragmentu układu pochylnia-belka widoczny jest na rysunku nr 13.



Rys. 13. Układ wciągający linkę z zamocowaną belką 1 – wałek, 2 – łożyska, 3 – sprzągło kłowe (w uproszczeniu), 4 – silnik krokowy (w uproszczeniu)

## 2.2 Realizacja zaprojektowanego obiektu sterowania

Przygotowana w programie SolidWorks konstrukcja układu kulka-belka posłużyła do zbudowania urządzenia. W niniejszym podrozdziale zostanie omówiona realizacja układu pod względem mechanicznym, elektrycznym i programistycznym.

### 2.2.1. Ujęcie mechaniczne

Konstrukcja obiektu sterowania została wykonana z drewna. Jako podstawy użyto płyty meblowej o wymiarach 120cm x 20cm x 2cm, do której przykręccono wykonane z plastiku nogi z regulacją długości. Posiadają one zastosowanie w przypadku postawienia układu na nierównej powierzchni. Natomiast ramiona podtrzymujące belkę zostały wykonane z drewna dębowego zgodnie z projektem. Drewno dębowe charakteryzuje się wysoką twardością i wytrzymałością mechaniczną, dlatego konstrukcja jest bardzo stabilna. Dębowe ramiona oraz plastikowe nogi przykręccono do podstawy wkrętami do drewna. Zostało to przedstawione na rysunku 14.



Rys. 14. Drewniana konstrukcja urządzenia

W wyfrezowanych otworach umieszczono łożyska kulkowe o symbolu 605 ZZ, które są widoczne na rysunku 15.



Rys. 15. Łożysko kulkowe 605 ZZ umieszczone w wyfrezowanym otworze

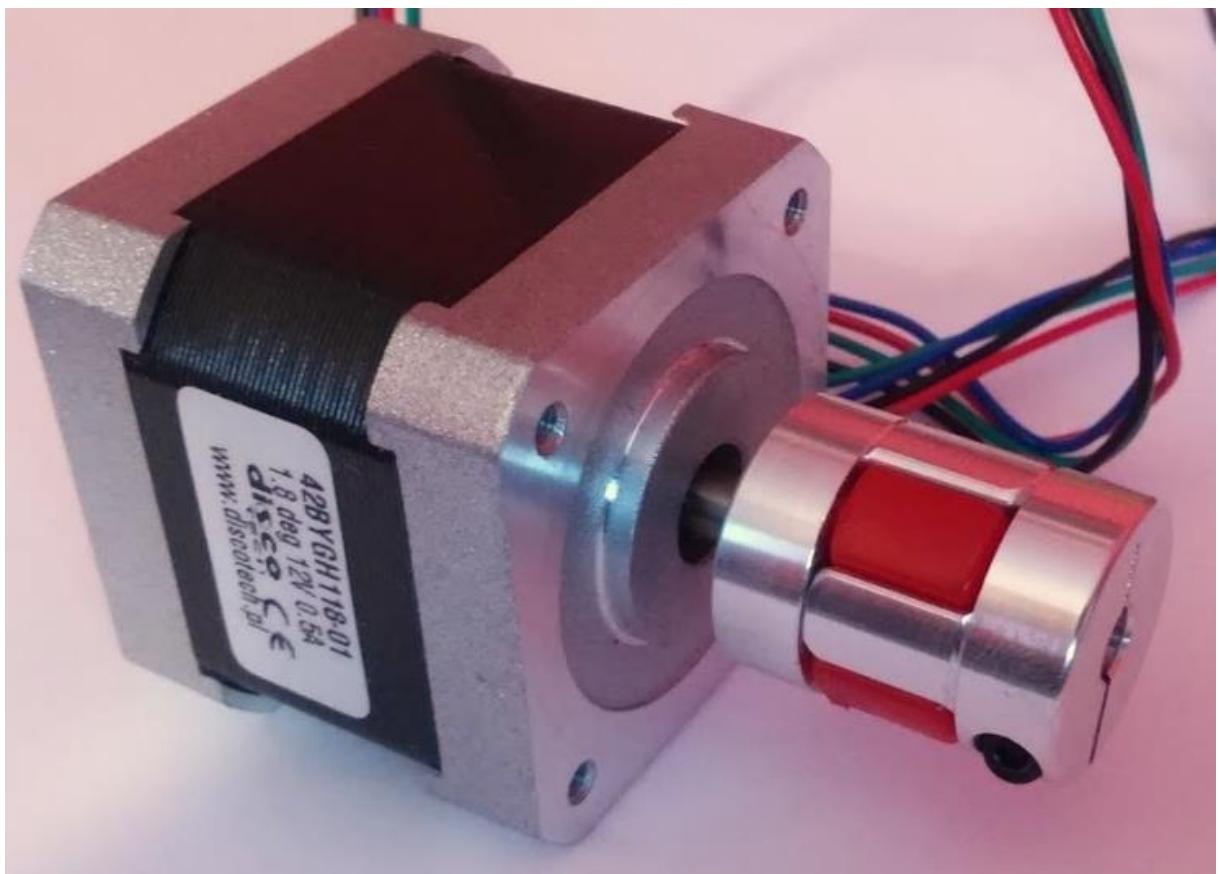
Łożysko kulkowe 605 ZZ jest łożyskiem jednorzędowym, nierożłącznym. Przystosowane jest do pracy z dużą prędkością obrotową oraz przenoszenia obciążzeń promieniowych. Składa się z pierścienia zewnętrznego, pierścienia wewnętrznego oraz znajdującego się pomiędzy pierścieniami koszyka z kulkami. Łożyska diametralnie zmniejszają tarcie podczas ruchu obrotowego wałka ułatwiając prace silnika [12].

Do połączenia wałka, na który będzie nawijana linka, z wałem wirnika silnika krokkowego użyto sprzęgło kłowego z wkładką elastyczną (Rysunek 16)



Rys. 16. Sprzęgło kłowe z wkładką elastyczną

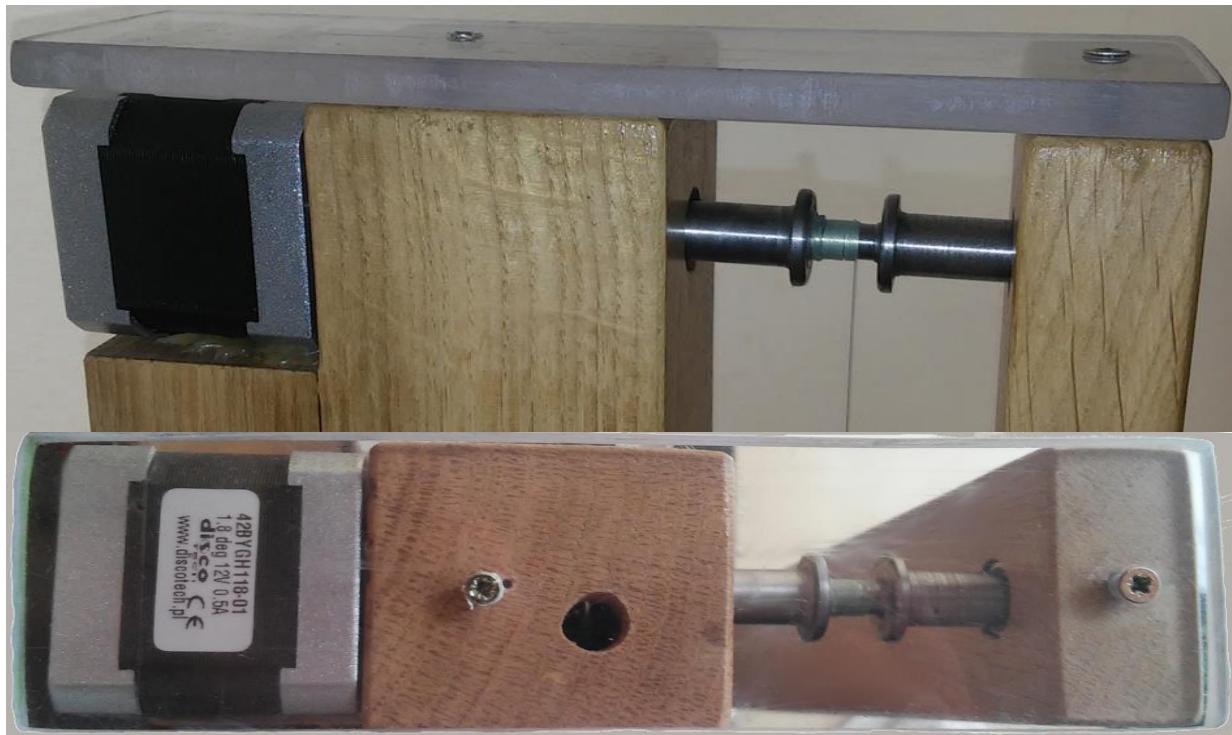
Sprzęgło nasunięto na wał wirnika silnika krokkowego, następnie zaciśnięto poprzez wkręcenie śruby imbusowej M3. Silnik z zamontowanym sprzęgiem zaprezentowano na rysunku 17.



Rys. 17. Silnik krokkowy bipolarny 42BYGH118-01 ze sprzęgiem kłowym

Tak przygotowany silnik zamontowano na ramieniu. Silnik został przymocowany klejem na ciepło, ponieważ wykonanie tak głębokich otworów, pod śruby do montażu silnika, które zakładał projekt, było zbyt trudne do wykonania. Spowodowane jest to twardością

i nierównomierną strukturą drewna dębowego. W ramieniu, do którego przyjmocowany jest silnik wywiercono otwór umożliwiający wsunięcie klucza imbusowego do śruby sprzęgła kłowego. Umożliwia to zaciśnięcie sprzęgła na wałku. Ramiona zostały połączone przy użyciu pleksi (rysunek 18).



Rys. 18. Mocowanie silnika oraz wałka z nawiniętą linką

Do linki nawiniętej na wałku przyjmocowano uchwyt wykonany zgodnie z rysunkiem technicznym (Załącznik 1). Jako materiału do wykonania uchwytu, użyto utwardzonej żywicy epoksydowej, powszechnie używanej w narzędziowniach. W górnej części uchwytu wkręceno śrubę oczkową M2,5, przez której oczko przewleczono linkę. Uchwyt zaprezentowano na rysunku 19.



Rys. 19. Uchwyt belki od strony silnika

Jako linki użyto żyłki wędkarskiej, o średnicy 0,35mm, której udźwig deklarowany przez producenta wynosi 12,7kg.



Rys. 20. Uchwyt zamontowany między ramionami

Uchwyt został zamontowany w odpowiedniej odległości, aby bez względu na kąt nachylenia belki, zawsze znajdował się pomiędzy ramionami, co obrazuje rysunek 20. Taki układ wyklucza ruch belki w innych płaszczyznach niż pionowa. Drewno zostało polakierowane, aby zmniejszyć tarcie podczas przesuwania uchwytu.

Omawiany uchwyt połączony został z belką za pomocą dwóch śrub. Belkę tą stanowi kątownik aluminiowy o wymiarach 15x15x2mm, widoczny na rysunku 21 [13].



Rys. 21. Kątownik aluminiowy

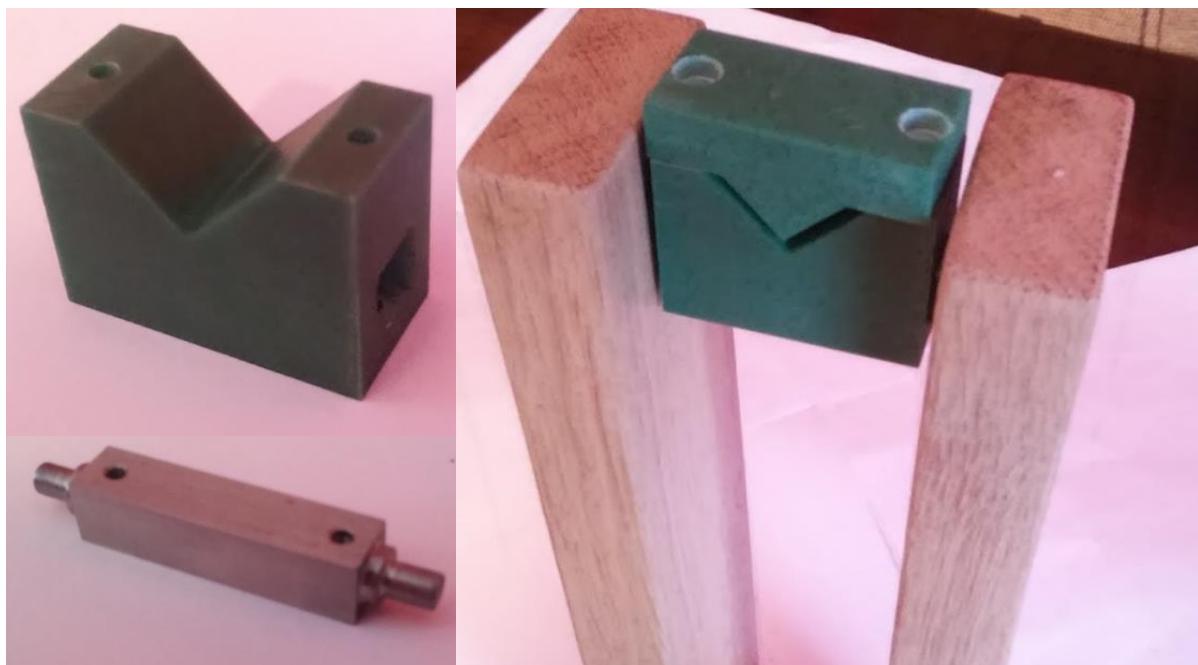
Kulka poruszająca się po belce to biała bila o średnicy 47,7mm, zaprezentowana na rysunku 22. Testowano również bilę o średnicy 60,3mm, jednakże jej masa powodowała znaczne wyginanie

belki. Ze względu na zastosowanie optycznych czujników odległości wybrano barwę białą, która najlepiej odbija światło.



Rys. 22. Biała bila na belce

Uchwyt trzymający belkę z drugiej strony również został wykonany z utwardzonej żywicy epoksydowej. Także składa się z dwóch części skręcanych ze sobą śrubami M3. W dolnej części uchwytu wykonano otwór w kształcie kwadratu, w który wsunięto element wykonany ze stalowego, kwadratowego pręta o wymiarach 10x10 mm. Pręt został przycięty na długość ok. 65mm, a jego końce przetoczono uzyskując fragmenty wałka o średnicy 5mm, które wsunięto w łożyska kulkowe. Jest to widoczne na rysunku nr 23.

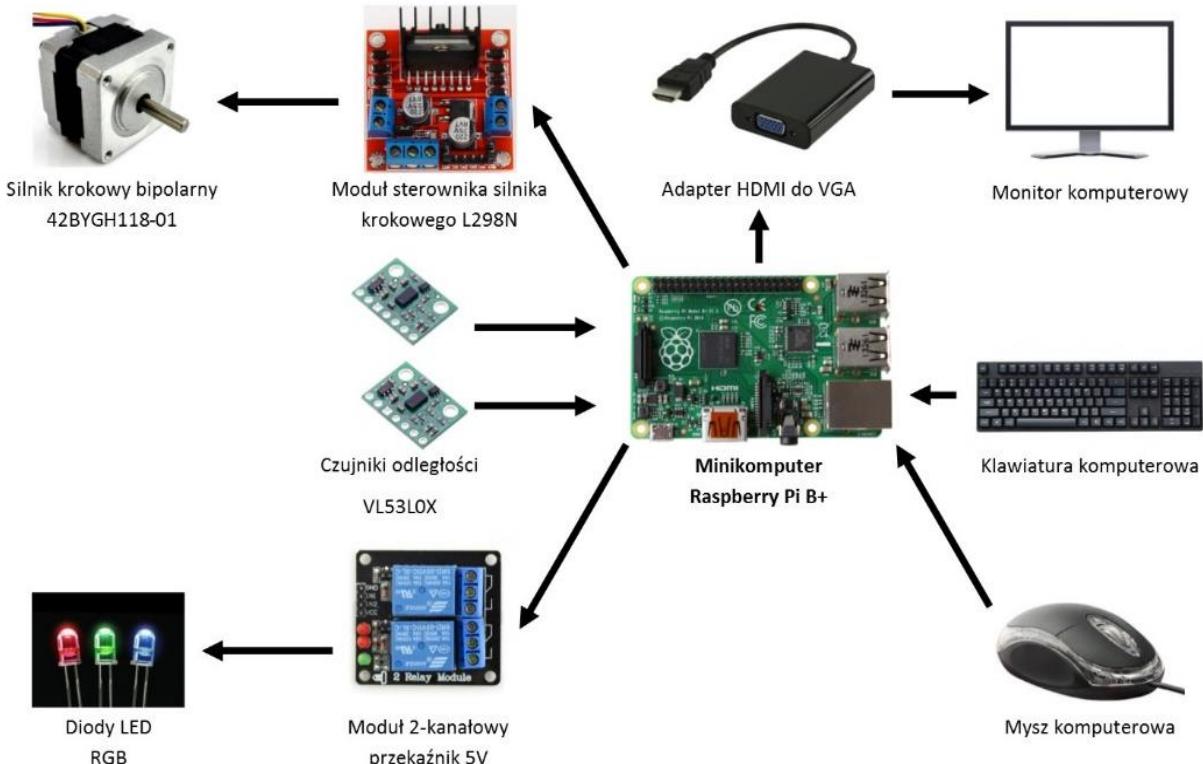


Rys. 23. Uchwyt belki

Tak skonstruowany uchwyt pozwala na zmianę pozycji pochylni w płaszczyźnie pionowej, na zasadzie zawiasu.

## 2.2.2 Ujęcie elektryczne

Urządzenie ma charakter budowy modułowej. Głównym modułem jest minikomputer Raspberry Pi model B+, odpowiedzialny za całą pracę urządzenia. Do niego podłączono pozostałe moduły rozszerzające możliwości minikomputera. Na rysunku 24 przedstawiono schemat budowy urządzenia, wraz z zaznaczonym strzałkami przepływem sygnałów pomiędzy podzespołami.



Rys. 24. Schemat budowy urządzenia

Do komunikacji użytkownika z urządzeniem służą klawiatura i mysz komputerowa, podłączone do gniazd USB. Wyniki pracy wyświetla monitor LCD podłączony poprzez adapter do wyjścia HDMI. Moduły podłączono do pinów GPIO minikomputera.

Problematyka budowy urządzenia w aspekcie elektronicznym składa się z omówienia następujących elementów:

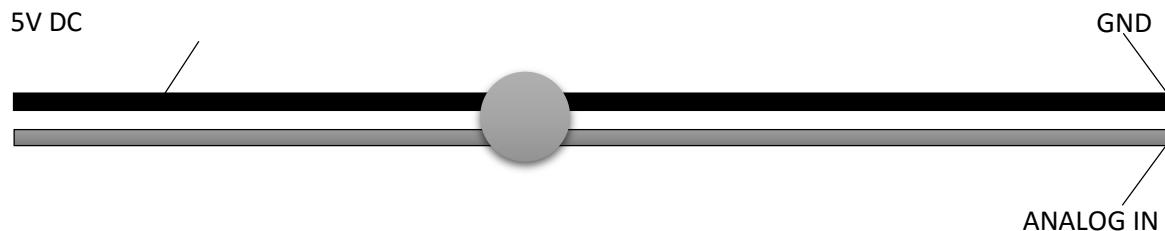
- pomiar pozycji kulki na belce
- pomiar nachylenia pochylni
- zmiana konta nachylenia belki
- sterowanie układem

ad. A. W trakcie budowy urządzenia testowano różne metody pomiaru pozycji kulki na belce. Były to budowa potencjometru liniowego, czujniki ultradźwiękowe, czujniki optyczne analogowe i czujniki optyczne cyfrowe.

Pierwsza metoda zakładała budowę potencjometru liniowego. Belka układu została zbudowana z 2 równolegle ułożonych stalowych prętów. Jeden został zaizolowany, następnie nawinięto na niego uzwojenia z drutu oporowego Kanthal typu D, o średnicy 0.20mm. Do końców uzwojenia przyłożono napięcie wynoszące 5V DC. Stalowa kulka stykała uzwojenie z drugim odizolowanym prętem. Zmierzono wartość napięcia na drugim pręcie w kilku pozycjach kulki. Na podstawie tych pomiarów można było znaleźć zależność między napięciem na pręcie, a pozycją kulki na belce. Rozwiążanie to przetestowano po zbudowaniu pierwszego prototypu, widocznego na rysunku 25. Jako mikrokontrolera użyto Arduino UNO, do którego podłączono belkę jak na schemacie przedstawionym na rysunku 26.



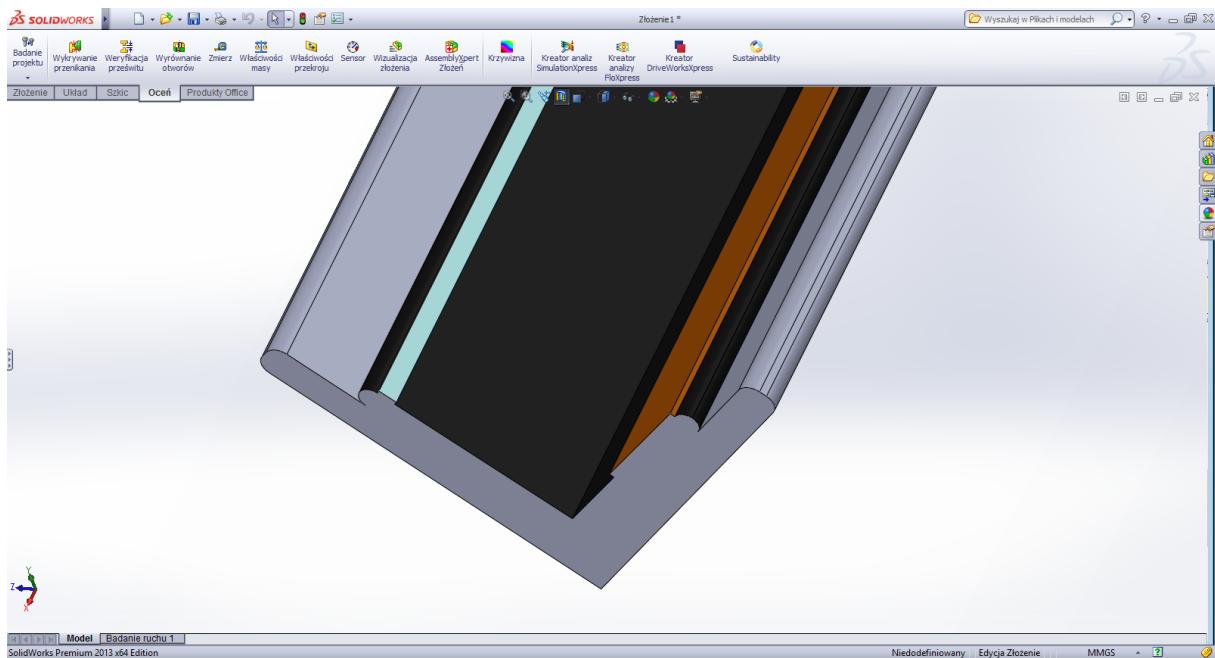
Rys. 25. Prototyp układu kulka belka do badania regulatora



Rys. 26. Schemat pomiaru pozycji kulki

Takie rozwiązanie pozwoliło na stabilny pomiar tylko w momencie, gdy kulka stała nieruchomo. Podczas gdy kulka zmieniała pozycję, tocząc się, podskakiwała na chropowatej powierzchni pręta z uzwojeniami. Powodowało to bardzo niestabilny pomiar, który uniemożliwiał poprawną pracę układu.

W celu uniknięcia podskakiwania kulki na zwojach, zmieniono stalowe pręty na kątownik aluminiowy. Do kątownika aluminiowego wklejono kątownik z tworzywa sztucznego izolując ścieżki elektryczne od kątownika aluminiowego. Do jednej ścianki przyklejono taśmę oporową Kanthal o wymiarach 0,80x0,15mm, natomiast do drugiej - taśmę miedzianą. Projekt belki z tymi taśmami widoczny jest na rysunku 27. Tak skonstruowany układ podłączono do Raspberry Pi poprzez moduł przetwornika A/C ADS1015.



Rys. 27. Projekt belki z taśmą oporową i miedzianą

Podczas wciągania i opuszczania belki, pochylnia wpadała w drgania, które powodowały chwilowe utraty styku kulki z taśmami. Ponadto rezystancja taśmy oporowej była kilkukrotnie mniejsza od użwojenia z drutu, dlatego zmiany mierzonego napięcia były zbyt małe, aby można było dokładnie wyznaczyć pozycję kulki.

Druga z przetestowanych metod pomiaru kulki na belce opierała się na czujniku ultradźwiękowym HC-SR04, który widoczny jest na rysunku 28.



Rys. 28. Czujnik ultradźwiękowy HC-SR04

Czujnik ten posiada 4 piny. Dwa piny do zasilania (VCC, GND) oraz dwa piny do pomiaru (TRIG, ECHO). W momencie, gdy na pinie TRIG pojawi się stan wysoki, głośnik wysyła sygnał ultradźwiękowy o częstotliwości 40Hz. Kiedy mikrofon rejestruje powracający sygnał, na pinie ECHO pojawia się stan wysoki. Pomiar polega na zmierzeniu czasu, który upływa od wysłania sygnału na pin TRIG, do momentu pojawienia się stanu wysokiego na pinie ECHO. Znając prędkość dźwięku w powietrzu, która wynosi 340m/s oraz czas, po którym sygnał ultradźwiękowy powrócił, można obliczyć jaką drogę pokonał do obiektu i z powrotem. Dzieląc wynik przez dwa, otrzymamy odległość obiektu od czujnika. Wzór matematyczny na te obliczenia widoczny jest poniżej.

$$d[m] = \frac{t[s] * 340[\frac{m}{s}]}{2} \quad (2.1)$$

d – dystans między czujnikiem, a obiektem

t – zmierzony czas między wysłanie sygnału, a odebraniem sygnału

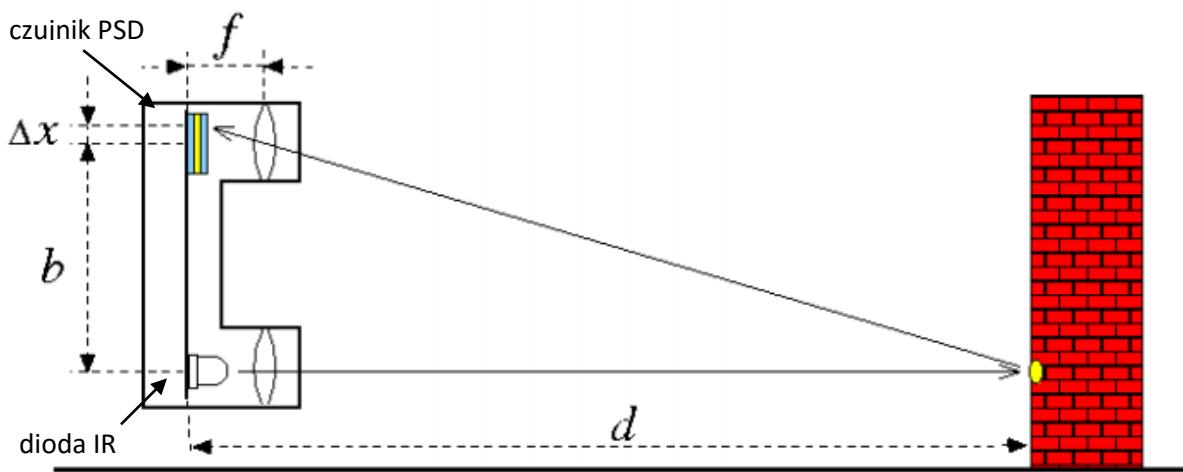
Czujnik ten nie wykrywał małej stalowej kulki, dlatego zastosowano większą kulę bilardową. Deklarowany przez producenta zakres pomiarowy wynosi 2-200cm. W trakcie testów zauważono, że czujnik mierzył poprawnie odległość w tym zakresie tylko w przypadku ustawienia dużej płaszczyzny prostopadle do czujnika. Natomiast pozycja kulki była mierzona poprawnie tylko w zakresie 5-25cm. Zastosowanie 2 czujników na krańcach belki dało zakres pomiarowy ok. 40cm. Mimo to pomiary nie były stabilne, ponieważ czujniki wysyłają sygnały w promieniu ok  $40^\circ$ , które odbijane są nie tylko przez kulę. Duża częstotliwość pomiarów powodowała odbieranie sygnałów ultradźwiękowych odbitych od obiektów z otoczenia, które nie zdążyły jeszcze wygasnąć. Dawało to błędne pomiary, tym samym uniemożliwiając pracę regulatora.

Trzecia metoda zakładała użycie analogowego czujnika optycznego Sharp GP2Y0A21YK0F, który został przedstawiony na rysunku 29.



Rys. 29. Analogowy czujnik optyczny Sharp GP2Y0A21YK0F

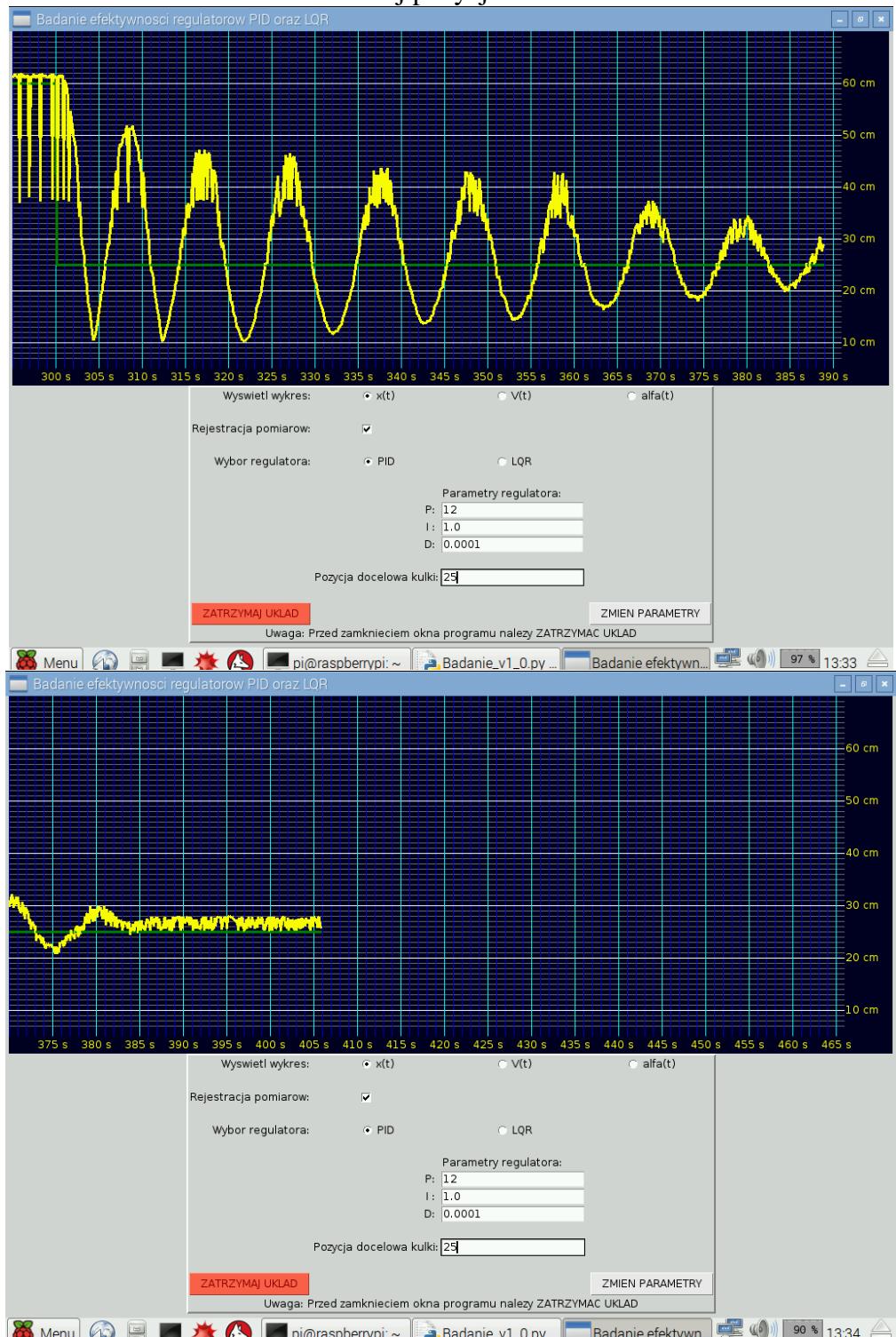
Czujnik ten posiada 3 piny. Dwa piny służą do zasilania (Vcc, GND), natomiast na trzecim (Vo) pojawia się napięcie o wartości zależnej od odległości między czujnikiem, a mierzonym obiektem. Minikomputer Raspberry Pi nie posiada wejść analogowych, dlatego napięcie na pinie Vo czujników mierzone było za pośrednictwem modułu przetwornika A/C ADS1015. Sharp GP2Y0A21YK0F jest to dalmierz triangulacyjny wykorzystujący czujnik PSD (Position Sensitive Device). Schemat pomiaru odległości dalmierzem na zasadzie triangulacji przedstawiony został na rysunku 30 [14].



Rys. 30. Schemat pomiaru odległości dalmierzem na zasadzie triangulacji

Za pomocą diody podczerwonej i soczewki, czujnik wysyła skupioną wiązkę światła, która odbita od obiektu częściowo wraca do drugiej soczewki, a następnie pada na czujnik PSD, pozwalający zmierzyć w którym miejscu padło światło. Według dokumentacji, czujnik powinien poprawnie mierzyć dystans w zakresie 10-80cm, jednak w trakcie testów stwierdzono, iż zakres w przypadku mierzenia odległości kuli bilardowej jest mniejszy oraz wynosi

około 10-35cm. Stąd na krańcach belki zamontowano dwa czujniki. Rysunek poniżej przedstawia przykładowy wykres pracy pozycji kulki w czasie pracy regulatora. Odczyty pozycji uzyskane z czujników nie zawsze były stabilne, co widać na rysunku 31, jednak regulator pozwalał na ustawianie kulki w oczekiwanej pozycji.



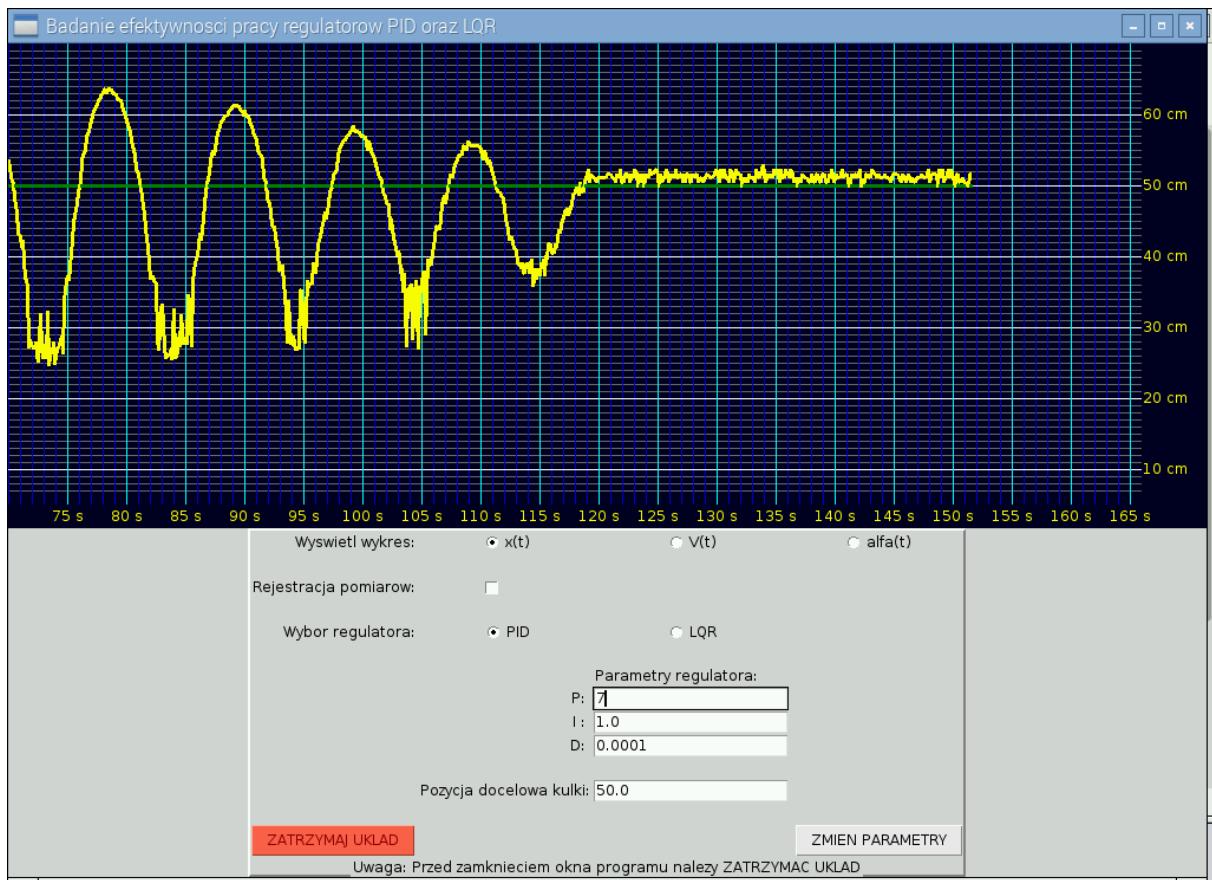
Rys. 31. Wykres pozycji kulki w czasie pracy układu - pomiar dalmierzami Sharp GP2Y0A21YK0F

Ostatnia metoda opierała się na czujniku optycznym Pololu VL53L0X, zaprezentowanym na rysunku 32.



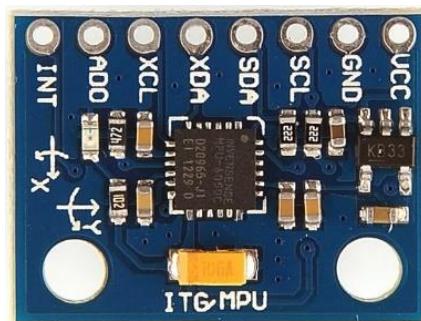
Rys. 32. Czujnik optyczny Pololu VL53L0X

Czujnik ten działa na zasadzie time-of-flight, tak jak czujnik ultradźwiękowy HC-SR04. Z tą różnicą, że tutaj mierzony jest czas między wysłaniem impulsu z lasera podczerwonego, a wykryciem impulsu przez detektor. Do komunikacji z komputerem Raspberry Pi wykorzystuje magistralę I2C. Według specyfikacji czujnik mierzy obiekty odległe do 200cm, jednak efektywny pomiar odległości białej kuli od czujnika wynosi około 35cm. W celu rozszerzenia zakresu pracy układu, użyto dwóch czujników, uzyskując dzięki temu pomiary na długości 70cm. Pomiary są bardziej stabilne niż w przypadku poprzednich czujników. Zakłócenia w pomiarach występują, gdy kulka znajduje się w okolicy połowy odległości między czujnikami. Oba dalmierze mają taki sam adres na magistrali I2C, aby otrzymać pomiar z obu należy naprzemiennie wyłączać jeden z nich podając stan wysoki na pin XSHUT. Poniżej przedstawiono przykładowy wykres pozycji kulki podczas pracy regulatora, gdy pomiary były prowadzone poprzez czujniki Pololu VL53L0X (rysunek 33).



Rys. 33. Wykres pozycji kulki w czasie pracy układu - pomiar dalmierzami Pololu VL53L0X

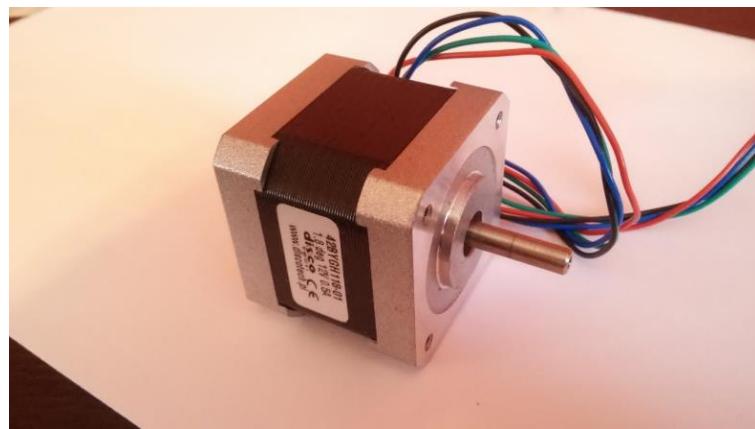
Ad. B. Drugi z aspektów elektronicznych dotyczy pomiaru kąta nachylenia belki. Do tego pomiaru użyto modułu MPU-6050, widocznego na rysunku 34.



Rys. 34. Moduł MPU-6050

MPU-6050 jest to 3 osiowy moduł żyroskopu z przyśpieszeniomierzem. Do komunikacji z kontrolerem wykorzystuje interfejs I2C. Moduł ten pracuje w zakresie pracy do 2500°/s. W trakcie testów zauważono, iż żyroskop przy większych drganiach wysyła błędne pomiary. Podczas pracy urządzenia belka drga nieustannie, przez co otrzymywane wyniki z żyroskopu były nieprzydatne. Problem pomiaru aktualnego kąta nachylenia pochylni rozwiązano poprzez zliczanie kroków silnika krokkowego.

Ad. C. Trzeci aspekt elektroniczny dotyczy zmiany kąta nachylenia pochylni. Aktorem, czyli członem wykonawczym, w układzie jest silnik bipolarny o symbolu 42BYGH118-01, widoczny na rysunku 35. Wybrane parametry silnika zawiera tabela nr 1 [15].



Rys. 35. Silnik bipolarny 42BYGH118-01

Tabela 1. Wybrane parametry silnika krokkowego, bipolarnego 42BYGH118-01

Napięcie znamionowe	Prąd znamionowy	Moment	Krok	Temperatura pracy
12V	0,5A	0,42Nm	1,8°±5%	0 - 40°C

Silnik połączony mechanicznie z wałkiem, nawija lub rozwija linkę, tym samym unosząc lub opuszczając jeden z końców pochylni. Zastosowanie silnika krokkowego pozwoliło na bardzo precyzyjne sterowanie nachyleniem belki, bo z dokładnością do 0,0027°. Wynik ten uzyskano dzięki działaniom matematycznym przedstawionym na rysunku 36.

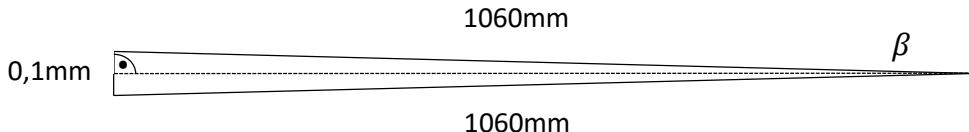
$$l = \frac{\alpha}{360^\circ} \pi d \quad (2.2)$$

$$l \approx \frac{1,8^\circ}{360^\circ} * 6mm * 3,14 \approx 0,1mm \quad (2.3)$$

l – długość łuku zakreślana przez wałek przy jednym kroku

$\alpha$  – kąt o jaki obraca się wirnik silnika w jednym kroku

d – średnica wałka, na który nawijana jest linka

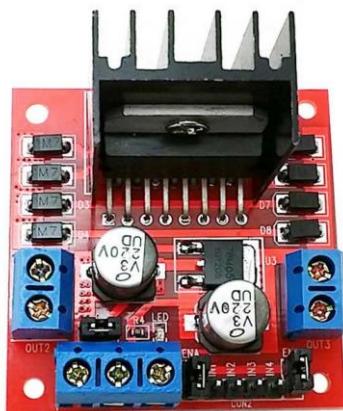


Rys. 36. Obliczenie dokładności sterowania nachyleniem belki

$$\sin\beta = \frac{1060mm}{0,05mm} \quad (2.4)$$

$$\beta \approx 0,027^\circ \quad (2.5)$$

Silnik podłączono do modułu dwukanałowego sterownika silników L298N (rysunek 37), którego wybrane parametry zamieszczono w tabeli nr 2 [16] [17].

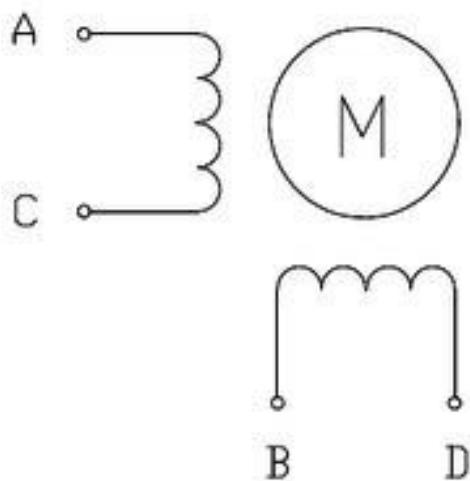


Rys. 37. L298N - dwukanałowy sterownik silników

Tabela 2. Parametry modułu sterownika L298N

Napięcie zasilania silnika	Zasilanie części logicznej	Maksymalny prąd wyjściowy na kanał
12V	5V	2A

Do sterownika podłączono osobne zasilanie robocze o napięciu 12V DC. Natomiast część logiczna zasilana jest z mikrokontrolera. Silnikiem krokowym steruje się wysyłając sygnały w odpowiedniej sekwencji, na 4 piny wejściowe sterownika. Sterownik następnie przepuszcza prąd roboczy przez uzwojenia silnika w tej samej kolejności. Rysunek 38 przedstawia schemat silnika krokowego bipolarnego, 4-ro przewodowego [18].



Rys. 38. Schemat silnika krokowego bipolarnego, 4-ro przewodowego

Ad. D. Ostatni aspekt elektroniczny dotyczy systemu sterowania. Za sterowanie układem odpowiedzialny jest minikomputer jednopłytkowy Raspberry Pi model B+, widoczny na rysunku 39 [19].



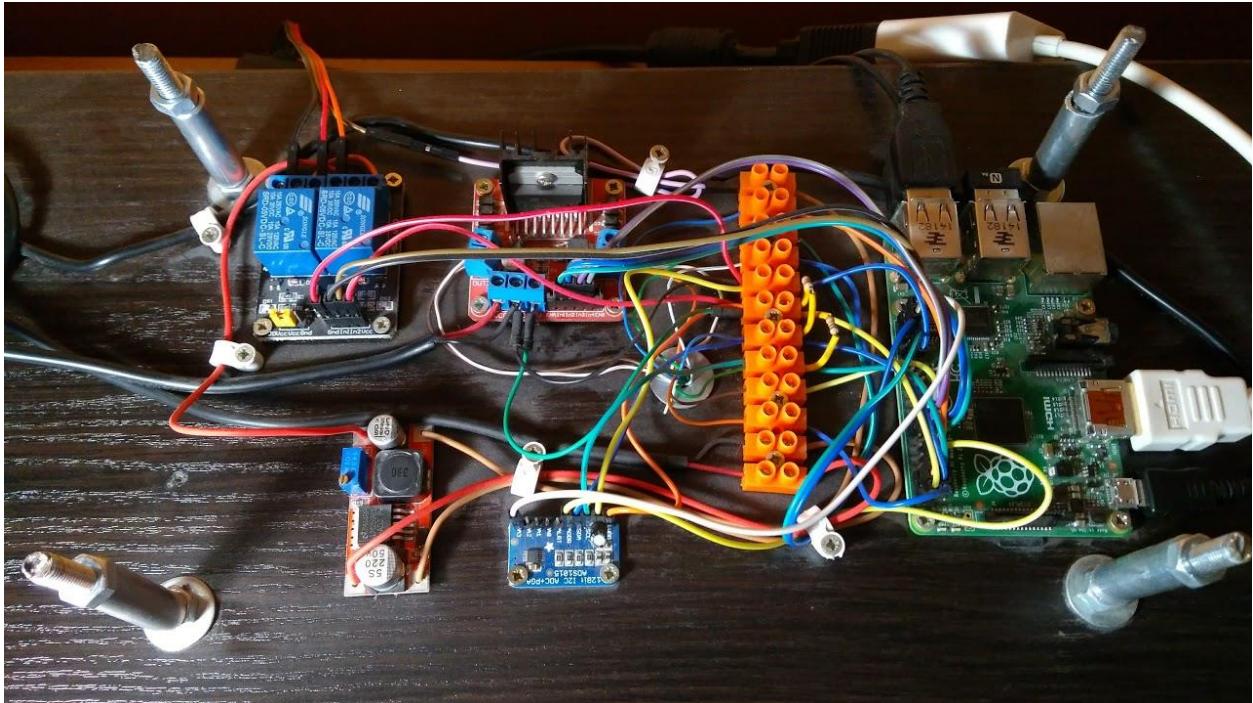
Rys. 39. Minikomputer Raspberry Pi B+

Raspberry Pi B+ wyposażony został w:

- procesor Broadcom CoS BCM2835 700 MHz
- pamięć RAM – 512MB
- gniazdo HDMI 1.4
- gniazdo zasilające microUSB
- złącze do dedykowanej kamery
- 4 porty USB
- 40 pinów GPIO
- gniazdo na karty microSD
- złącze jack 3,5mm
- gniazdo Ethernet [20].

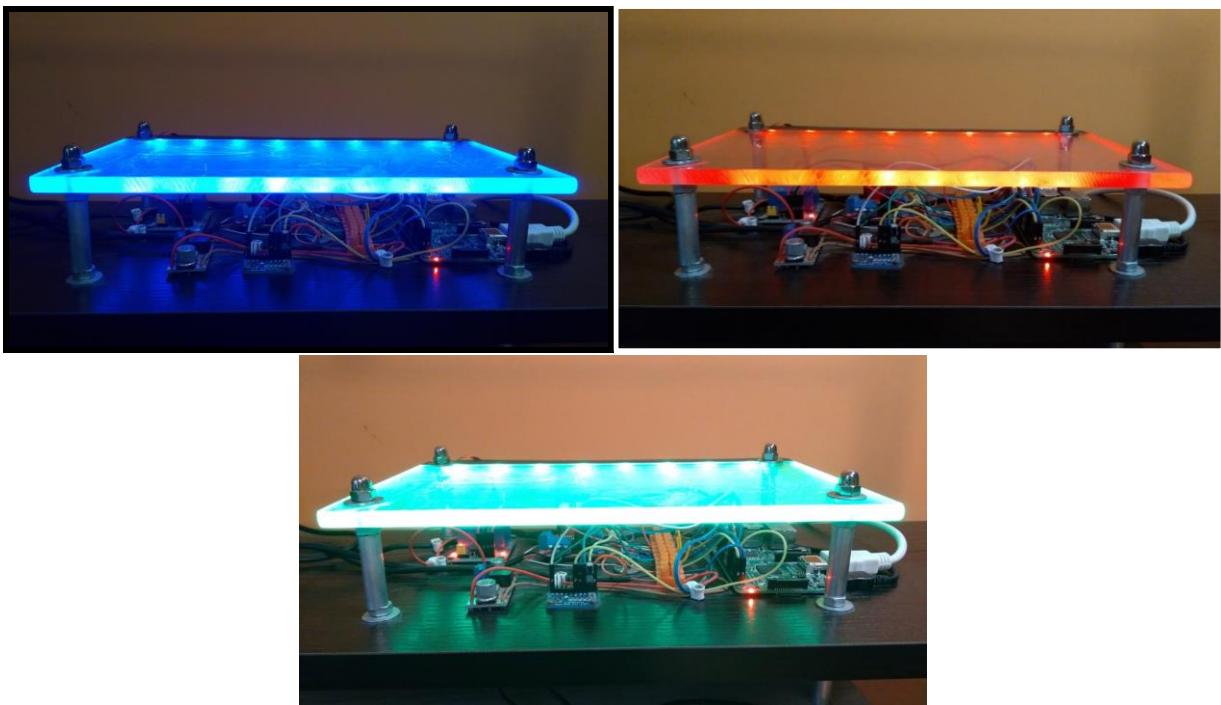
Na komputerze zainstalowano system operacyjny Raspbian. Jest to wolny i otwarty system oparty na jednej z dystrybucji Linuksa o nazwie Debian. System został specjalnie zoptymalizowany do działania na Raspberry Pi [21].

Wszystkie komponenty połączono przewodami połączeniowymi zakończonymi złączem raster 2,54mm oraz przewodami sygnalowymi tzw. skrętką. W załączniku nr 3 zamieszczono pełen schemat elektryczny. Natomiast na rysunku 40 przedstawiono połączone moduły urządzenia.



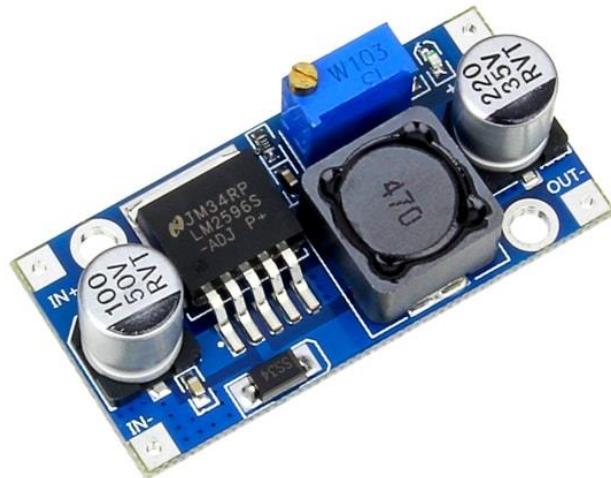
Rys. 40. Połączone moduły urządzenia

Na rysunku powyżej widoczne są również cztery szpilki, na których zamocowano fragment przeźroczystej płyty pleksi. Pełni ona rolę ochronną i sygnalizacyjną. Płyta o grubości 8mm chroni komponenty elektroniczne przed uszkodzeniami mechanicznymi, które mogłyby nastąpić, gdyby kula wypadła z belki zawieszonej powyżej. Ponadto w płycie nawiercono osiem otworów, w które wsunięto diody RGB 5mm. Diody podświetlają płytę w trzech kolorach. Barwa niebieska oznacza, że program aktualnie nie jest uruchomiony. Płyta podświetloną kolorem czerwonym informuje, że układ pracuje, a kula nie znajduje się w zadanej pozycji. Natomiast barwa zielona sygnalizuje, że kula znajduje się blisko pozycji zadanej. Na rysunku 41 pokazano zainstalowaną płytę pleksi podświetloną w trzech wariantach.



Rys. 41. Płyta ochronna z sygnalizacją świetlną

Zastosowane diody RGB podłączone zostały poprzez osobne źródło zasilania 5V, które zostało obniżone do ok. 3,25V za pomocą modułu z przetwornicą impulsową step-down LM2596. Moduł ten umożliwia płynną regulację napięcia wyjściowego w zakresie od 3,2V do 35V. Natomiast maksymalne napięcie wejściowe wynosi 40V. Przetwornica może działać pracą ciągłą przy maksymalnym natężeniu prądu 2A. Moduł ten pokazano na rysunku nr 42 [22] [23].



Rys. 42. Moduł przetwornicy step-down LM2596

Zmiana barwy podświetlenia odbywa się za pośrednictwem dwukanałowego modułu przekaźników RM5. Moduł ten zbudowany jest z dwóch przekaźników SRD-05 z cewką zasilaną napięciem 5V. Maksymalne napięcie robocze bezpieczne dla urządzenia wynosi 250V AC, 110V DC. Natomiast maksymalne natężenie prądu roboczego to 10A. Moduł ten zaprezentowano na rysunku nr 43 [24] [25].



Rys. 43. Dwukanalowy moduł przekaźników RM5

### 2.2.3 Ujęcie programistyczne

Urządzenie zostało zaprogramowane w języku programowania Python. Jest to język skryptowy wysokiego poziomu. Program napisany w Pythonie jest bardzo przejrzysty, dzięki czemu jest łatwy do nauczenia. Cechą ta powoduje, że język ten staje się coraz bardziej popularny [26]. Język programowania Python może wyróżniać to, iż kod programu można pisać w dowolnym edytorze tekstu, ponieważ program nie musi być kompilowany. Wykonywany jest bezpośrednio przez interpretator. Ponadto nie ma potrzeby definiowania typu zmiennej. Zmienne tworzą się automatycznie w momencie przypisania im wartości. W zależności od wartości deklarowany jest domyślny typ zmiennej. W systemie operacyjnym Raspbian preinstalowany jest interpretator Pythona oraz edytory skryptów ułatwiające programowanie, takie jak Python IDLE. Na rysunku 44 zaprezentowano fragment kodu zapisany w Python IDLE.

```

*Badanie_v1_5.py - /home/pi/Badanie_regulatorow/Badanie_v1_5.py (2.7.9)*
File Edit Format Run Options Windows Help
print Kat_silnik
GPIO.cleanup()

def poziomuj(self):
    self.start_ukladu.grid()
    global kat_silnik
    kat_silnik = 0
    #ZAPISYWANIE DO PLIKU
def rejestr(self):
    global rejestracja
    rejestracja = self.rejestracja.get()
    if rejestracja == 1:
        print "reje=", rejestracja
        global data
        data = time.strftime("pomiary/%H_%M_%S____%d_%m.txt")
        try:
            plik = open(data, 'a')
            plik.write("Badanie regulatora PID \n")
            plik.write("t(s) x(cm) V(cm/s) alfa(*) wyjscie_PID(*) \n")
            plik.close()
        except:
            pass

def przelacznik_wyswietl(self):
    global wyswietl
    wyswietl = self.zmienna_wyswietl.get()

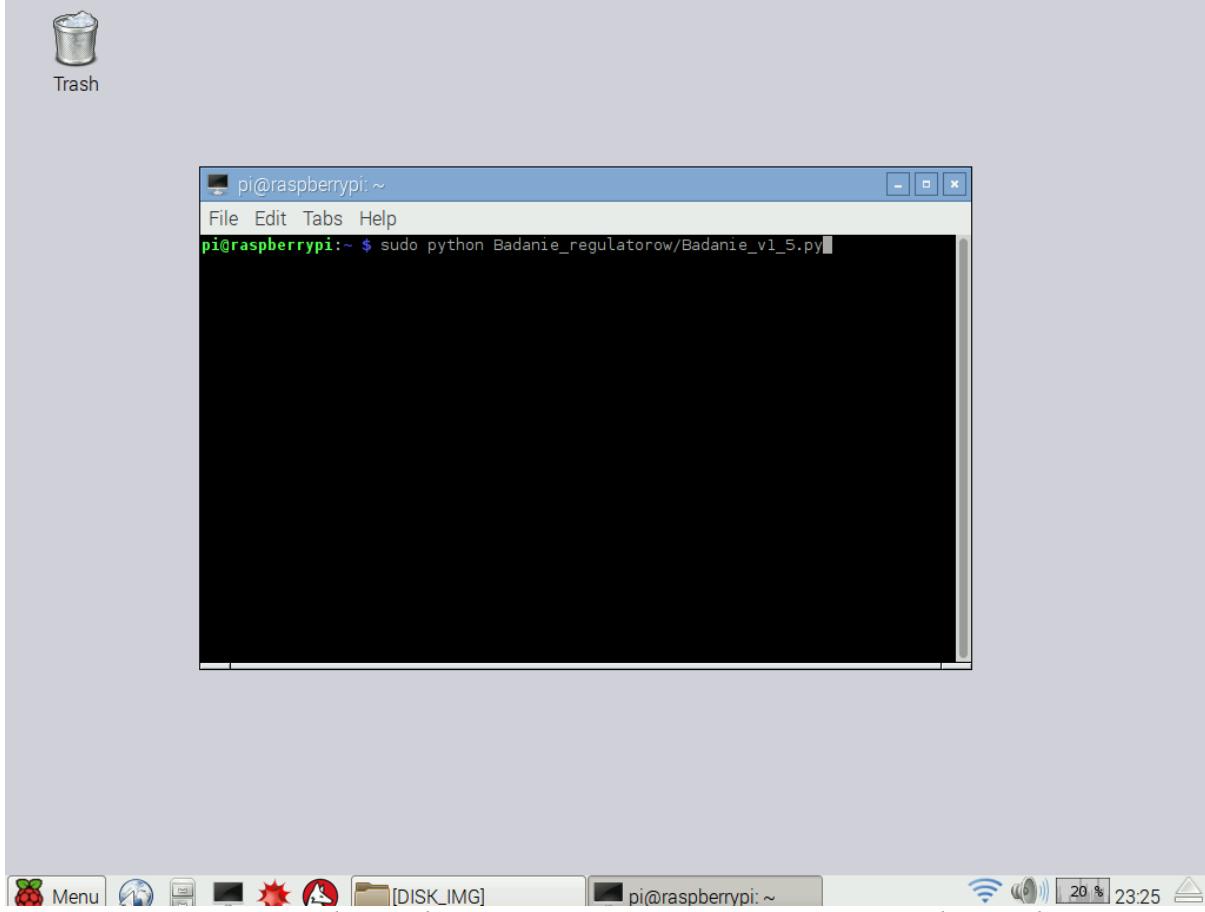
    print "abcs"
    if wyswietl == 1:
        gf.delete("all")

    y = 7
    while y < 70:
        linie_pozieme = y * 6
        linie_pozime = linie_pozieme - 420
        linie_pozome = -linie_pozime
        gf.create_line(0, linie_pozome, 960, linie_pozome,
                      fill="dimgray", width=1)
        y = y + 1
    x = 0
    while x <= 950:
        gf.create_line(x, 0, x, 390, fill="mediumblue", width=1)
        x = x + 1

```

Rys. 44. Kod napisany w edytorze skryptów – Python IDLE

Pliki napisane w Pythonie mają rozszerzenie .py. Program uruchamia się wpisując w terminalu komendę: „python ścieżka\_do\_pliku”. Uruchamiając programy, które korzystają z pinów GPIO należy użyć komendy: „sudo python ścieżka\_do\_pliku”, wtedy program jest uruchamiany przez administratora, tzw. roota, dzięki czemu ma dostęp do wszystkich zasobów systemu. Na rysunku 45 pokazano komendę w terminalu, uruchamiającą program do badania regulatora PID.



Rys. 45. Komenda uruchamiająca program, wpisana w terminalu Raspberry Pi

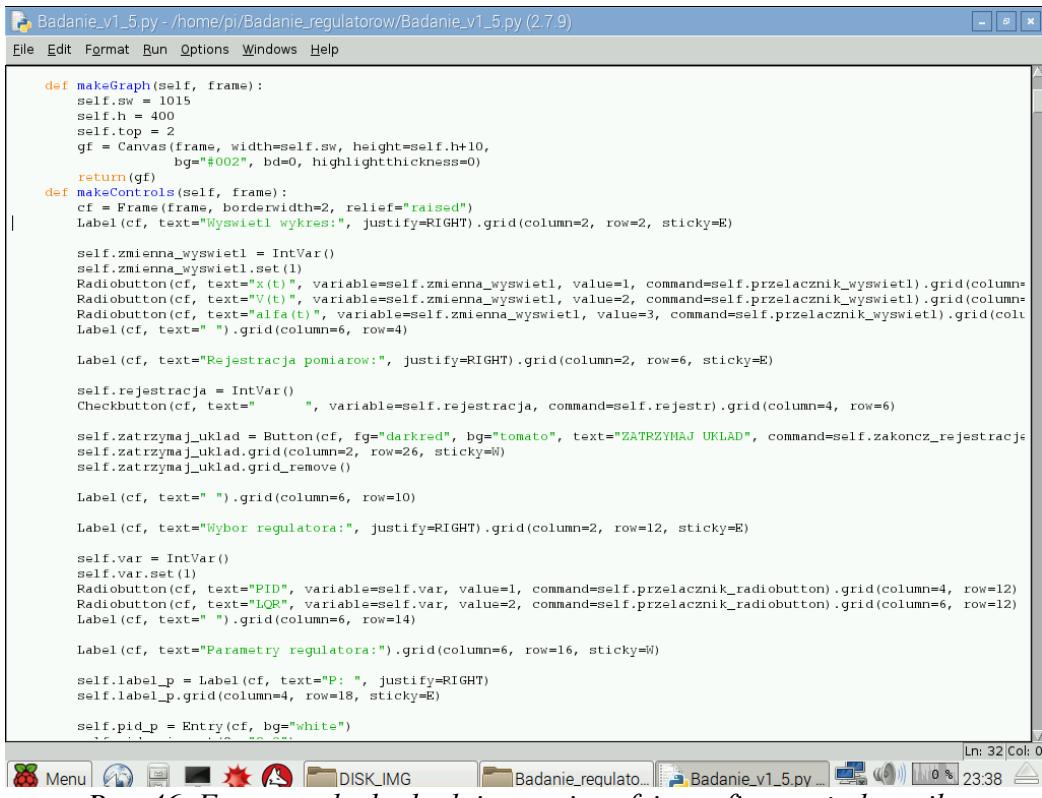
Program, który obsługuje układ do badania regulatora składa się z kilku funkcji:

- A. Wyświetlanie interfejsu graficznego użytkownika
- B. Wyświetlanie wykresu w czasie rzeczywistym
- C. Zapisywanie pomiarów do pliku tekstowego
- D. Odczyt pozycji kulki na pochylni
- E. Obliczenia regulatora PID
- F. Zmiana nachylenia belki

Ad. A. Pierwsza z funkcji to wyświetlanie interfejsu graficznego użytkownika. Do jego budowy wykorzystano wbudowaną w Pythona bibliotekę Tkinter. Biblioteka ta posiada standardowe widżety, między innymi: etykiety, przyciski, pola tekstowe, pola graficzne, listy, ramki.

Tworzenie okna dialogowego przy użyciu Tkinter polega na przywoływaniu widżetu komendą, następnie w nawiasie można podać parametry między innymi: kolor, tekst, wyrównanie do krawędzi, zakotwiczenie w ramce. Możliwości Tkintera, jak i samego Pythona można rozszerzać dodając dodatkowe biblioteki, również napisane w innych językach programowania.

Widżety można porządkować w kolumny i wiersze. Służy do tego komenda „grid()”, którą można dopisać po parametrach. W nawiasie należy wpisać w którym wierszu i kolumnie element ma zostać zakotwiczony. Na rysunku nr 46 zaprezentowano fragment kodu, za pomocą którego zbudowano interfejs graficzny użytkownika.



```

Badanie_v1_5.py - /home/pi/Badanie_re regulatorow/Badanie_v1_5.py (2.7.9)
File Edit Format Run Options Windows Help

def makeGraph(self, frame):
    self.sw = 1015
    self.h = 400
    self.top = 2
    gf = Canvas(frame, width=self.sw, height=self.h+10,
                bg="#002", bd=0, highlightthickness=0)
    return(gf)

def makeControls(self, frame):
    cf = Frame(frame, borderwidth=2, relief="raised")
    Label(cf, text="Wyswietl wykres:", justify=RIGHT).grid(column=2, row=2, sticky=E)
    self.zmienna_wyswietl = IntVar()
    self.zmienna_wyswietl.set(1)
    Radiobutton(cf, text="x(t)", variable=self.zmienna_wyswietl, value=1, command=self.przelacznik_wyswietl).grid(column=3)
    Radiobutton(cf, text="V(t)", variable=self.zmienna_wyswietl, value=2, command=self.przelacznik_wyswietl).grid(column=4)
    Radiobutton(cf, text="alfa(t)", variable=self.zmienna_wyswietl, value=3, command=self.przelacznik_wyswietl).grid(column=5)
    Label(cf, text=" ").grid(column=6, row=4)

    Label(cf, text="Rejestracja pomiarow:", justify=RIGHT).grid(column=2, row=6, sticky=E)
    self.rejestracja = IntVar()
    Checkbutton(cf, text=" ", variable=self.rejestracja, command=self.rejestr).grid(column=4, row=6)

    self.zatrzymaj_uklad = Button(cf, fg="darkred", bg="tomato", text="ZATRZYMAJ UKLAD", command=self.zakoncz_rejestracji)
    self.zatrzymaj_uklad.grid(column=2, row=26, sticky=W)
    self.zatrzymaj_uklad.grid_remove()

    Label(cf, text=" ").grid(column=6, row=10)

    Label(cf, text="Wybor regulatora:", justify=RIGHT).grid(column=2, row=12, sticky=E)
    self.var = IntVar()
    self.var.set(1)
    Radiobutton(cf, text="PID", variable=self.var, value=1, command=self.przelacznik_radiobutton).grid(column=4, row=12)
    Radiobutton(cf, text="LQR", variable=self.var, value=2, command=self.przelacznik_radiobutton).grid(column=6, row=12)
    Label(cf, text=" ").grid(column=6, row=14)

    Label(cf, text="Parametry regulatora:").grid(column=6, row=16, sticky=W)
    self.label_p = Label(cf, text="P: ", justify=RIGHT)
    self.label_p.grid(column=4, row=18, sticky=E)
    self.pid_p = Entry(cf, bg="white")
    self.pid_p.grid(column=6, row=18, sticky=W)

    self.label_I = Label(cf, text="I: ")
    self.label_I.grid(column=4, row=19, sticky=E)
    self.pid_I = Entry(cf, bg="white")
    self.pid_I.grid(column=6, row=19, sticky=W)

    self.label_D = Label(cf, text="D: ")
    self.label_D.grid(column=4, row=20, sticky=E)
    self.pid_D = Entry(cf, bg="white")
    self.pid_D.grid(column=6, row=20, sticky=W)

    Label(cf, text="Pozycja docelowa kulk:").grid(column=2, row=22, sticky=E)
    self.pozycja_kulka = Entry(cf, bg="white")
    self.pozycja_kulka.grid(column=4, row=22, sticky=W)

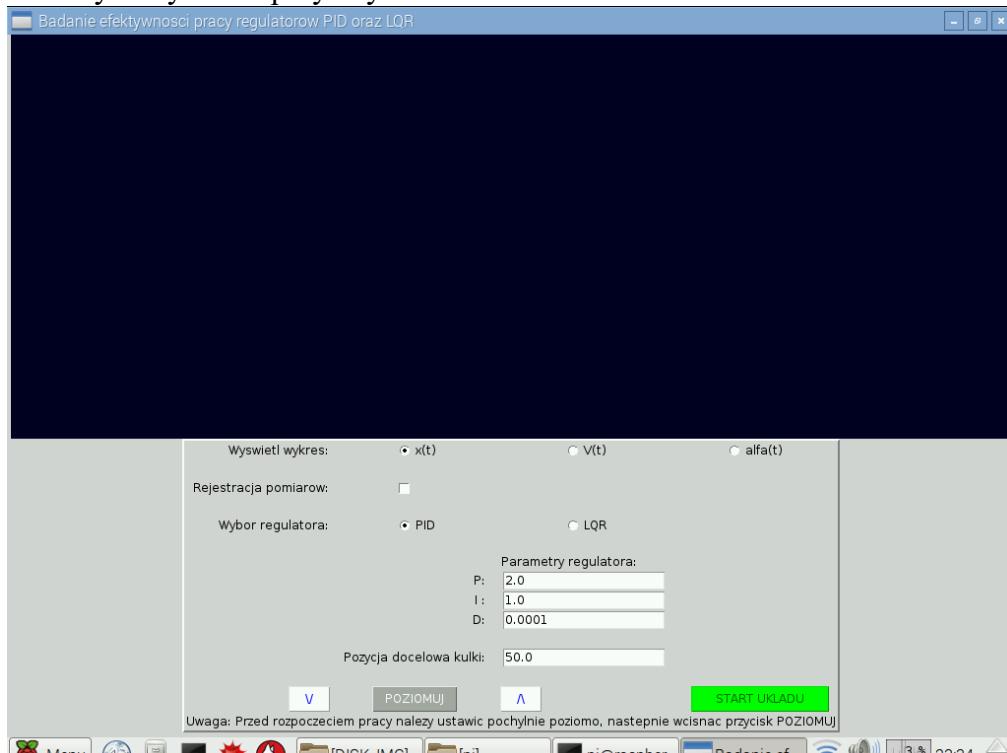
    self.v = Button(cf, text="V", bg="white", fg="blue", command=self.v)
    self.v.grid(column=3, row=24, sticky=W)
    self.POZIOMUJ = Button(cf, text="POZIOMUJ", bg="white", fg="blue", command=self.POZIOMUJ)
    self.POZIOMUJ.grid(column=4, row=24, sticky=W)
    self.a = Button(cf, text="A", bg="white", fg="blue", command=self.a)
    self.a.grid(column=5, row=24, sticky=W)
    self.START_UKLAU = Button(cf, text="START UKLAU", bg="green", fg="white", command=self.START_UKLAU)
    self.START_UKLAU.grid(column=6, row=24, sticky=W)

    Uwaga: Przed rozpoczęciem pracy należy ustawić pochyłość poziomo, następnie wcisnąć przycisk POZIOMUJ

```

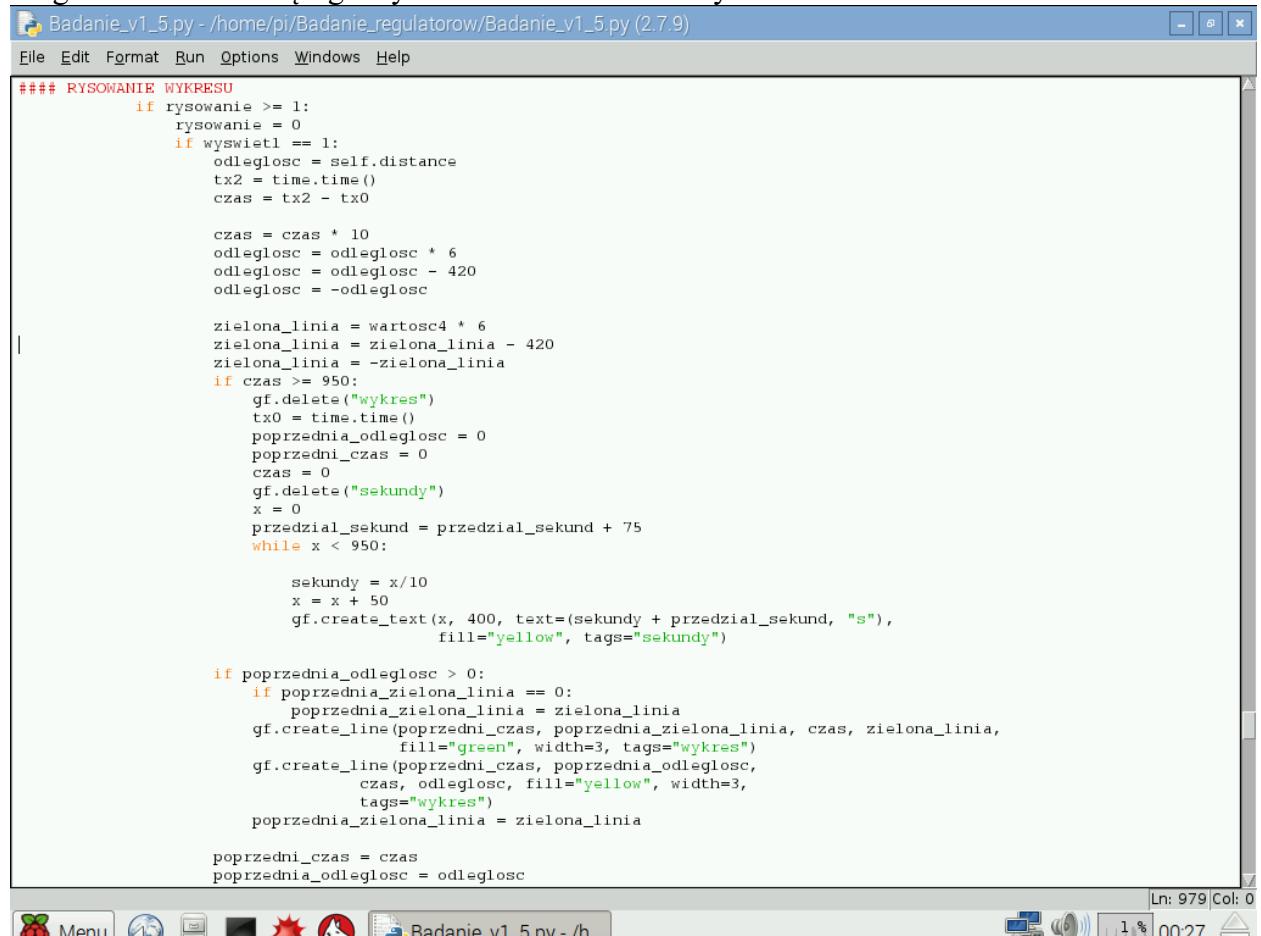
Rys. 46. Fragment kodu, budującego interfejs graficzny użytkownika

Elementy interfejsu, przez które użytkownik wprowadza dane (przyciski, pola do wpisania tekstu, listy wyboru, itp.) oraz elementy wyświetlające informacje użytkownikowi (etykiety, pola graficzne, itp.) muszą być zdefiniowane jako zmienne. Pozwoli to na dostęp do tych widżetów w innych fragmentach kodu. Na rysunku 47 pokazano interfejs graficzny zaprogramowany w Pythonie przy użyciu biblioteki Tkinter.



Rys. 47. Interfejs graficzny zaprogramowany w języku Python przy użyciu biblioteki Tkinter  
Interfejs składa się z kilku przełączników, przycisków oraz pól, w których można wpisać wartość nastaw regulatora.

Ad. B. Druga funkcja to wyświetlanie wykresu w czasie rzeczywistym. W górnej części okna, znajduje się pole, na którym po uruchomieniu układu wyświetlany jest wykres. W bibliotece Tkinter brakuje narzędzi, które pozwalałoby na tworzenie wykresów w czasie rzeczywistym, dlatego użyto widżetu „Canvas”. Narzędzie to rysuje na ekranie pole, na którym za pomocą odpowiednich komend można rysować proste figury geometryczne. Głównie korzystano z komendy tworzącej odcinki. Jako parametry wpisano zmienne których wartości określają współrzędne końców odcinka oraz parametry wizualne, takie jak kolor i grubość linii. Fragment kodu tworzącego wykres zamieszczono na rysunku 48.



```

#### RYSOWANIE WYKRESU
if rysowanie >= 1:
    rysowanie = 0
if wyswietl == 1:
    odlegosc = self.distance
    tx2 = time.time()
    czas = tx2 - tx0

    czas = czas * 10
    odlegosc = odlegosc * 6
    odlegosc = odlegosc - 420
    odlegosc = -odlegosc

    zielona_linia = wartosc4 * 6
    zielona_linia = zielona_linia - 420
    zielona_linia = -zielona_linia
    if czas >= 950:
        gf.delete("wykres")
        tx0 = time.time()
        poprzednia_odlegosc = 0
        poprzedni_czas = 0
        czas = 0
        gf.delete("sekundy")
        x = 0
        przedzial_sekund = przedzial_sekund + 75
        while x < 950:

            sekundy = x/10
            x = x + 50
            gf.create_text(x, 400, text=(sekundy + przedzial_sekund, "s"),
                           fill="yellow", tags="sekundy")

        if poprzednia_odlegosc > 0:
            if poprzednia_zielona_linia == 0:
                poprzednia_zielona_linia = zielona_linia
            gf.create_line(poprzedni_czas, poprzednia_zielona_linia, czas, zielona_linia,
                           fill="green", width=3, tags="wykres")
            gf.create_line(poprzedni_czas, poprzednia_odlegosc,
                           czas, odlegosc, fill="yellow", width=3,
                           tags="wykres")
            poprzednia_zielona_linia = zielona_linia
        poprzedni_czas = czas
        poprzednia_odlegosc = odlegosc

```

Rys. 48. Fragment kodu wykonującego rysowanie wykresu

Ad. C. Trzecia funkcja to zapisywanie pomiarów do pliku tekstowego. Umożliwia ona zapisanie takich informacji, jak: czas od startu pomiaru, pozycję kulki na belce, kąt nachylenia belki oraz wartość na wyjściu regulatora PID.

Funkcję można uruchomić zaznaczając w interfejsie graficznym okienko: „Rejestracja pomiarów”. Plik z zarejestrowanymi pomiarami można zimportować np. do arkusza kalkulacyjnego, co ułatwia opracowanie wyników i analizę pomiarów. Fragment pliku z zarejestrowanym pomiarem przedstawia rysunek 49.

```

<10_49_48__16_07.txt>
File Edit Search Options Help
Badanie regulatora PID
t(s) x(cm) V(cm/s) alfa(*) wyjscie_PID(*)
Badanie regulatora PID
P=6.0 I=0.0 D=0.0 pozycja_docelowa=25.0

t(s) x(cm) V(cm/s) alfa(*) wyjscie_PID(*)
0.926 55.3 5.968 0.0 -7.026
1.082 56.6 -4197.832 -4.135 -7.327
1.243 58.0 -127.889 -8.348 -7.652
1.401 59.0 -81.197 -4.406 -7.884
1.562 60.1 8.503 -8.618 -8.139
1.723 60.9 74.839 -4.792 -8.324
1.881 61.2 98.51 -8.773 -8.394
2.039 61.5 131.299 -4.638 -8.463
2.199 61.5 123.428 -8.502 -8.463
2.36 61.2 100.12 -8.657 -8.394
2.524 61.4 78.816 -4.212 -8.44
2.683 61.3 103.978 -8.116 -8.417
2.844 60.7 55.13 -12.56 -8.278
3.004 59.6 -30.601 -8.309 -8.023
3.163 58.4 -132.607 -3.826 -7.745
3.321 57.3 -217.715 -7.961 -7.49
3.485 56.8 -271.484 -5.874 -7.374
3.529 56.0 -321.543 -7.613 -7.188
3.697 54.3 -479.521 -3.362 -6.794
3.802 53.2 -543.928 -5.024 -6.539
3.906 52.4 -662.855 -7.497 -6.353
4.008 51.3 -717.912 -5.681 -6.098
4.11 50.7 -799.839 -7.729 -5.959
4.279 49.2 -889.951 -4.096 -5.611
4.382 48.9 -913.569 -6.338 -5.542
4.487 47.1 -1115.796 -4.56 -5.124
4.591 46.5 -1151.712 -6.145 -4.985
4.695 45.3 -1277.472 -4.56 -4.707
4.798 45.4 -1147.033 -6.415 -4.73
4.903 44.3 -567.663 -4.106 -4.475

```

Rys. 49. Fragment pliku z zarejestrowanym pomiarem

Ad. D. Kolejna funkcja to odczyt pozycji kulki na pochylni. Odczyt odległości obiektu od czujników odbywa się poprzez interfejs I2C. Producent modułów czujników (Pololu VL53L0X) udostępnił bibliotekę obsługiwany przez Raspberry Pi. Przed skorzystaniem z biblioteki konieczna jest konfiguracja. Należy wybrać tryb pracy oraz wskazać na których pinach podłączono czujniki. Funkcje zawarte w bibliotece zwracają wartość, która powinna odpowiadać odległość obiektu od czujnika. W trakcie testów zauważono jednak, iż otrzymywane wyniki znaczco różnią się od rzeczywistej pozycji kulki. W celu skorygowania wyników, przesuwano kulkę co 5cm, następnie sprawdzano odczyt z czujników. Wyniki pomiaru zaprezentowano w tabeli 3. Wyniki pomiarów, gdy kulka znajdowała się w odległości większej niż 35cm były niestabilne, dlatego brano pod uwagę wyniki zaznaczone w tabeli szarym kolorem.

Tabela 3. Pomiary wykonane czujnikami Pololu VL53L0X

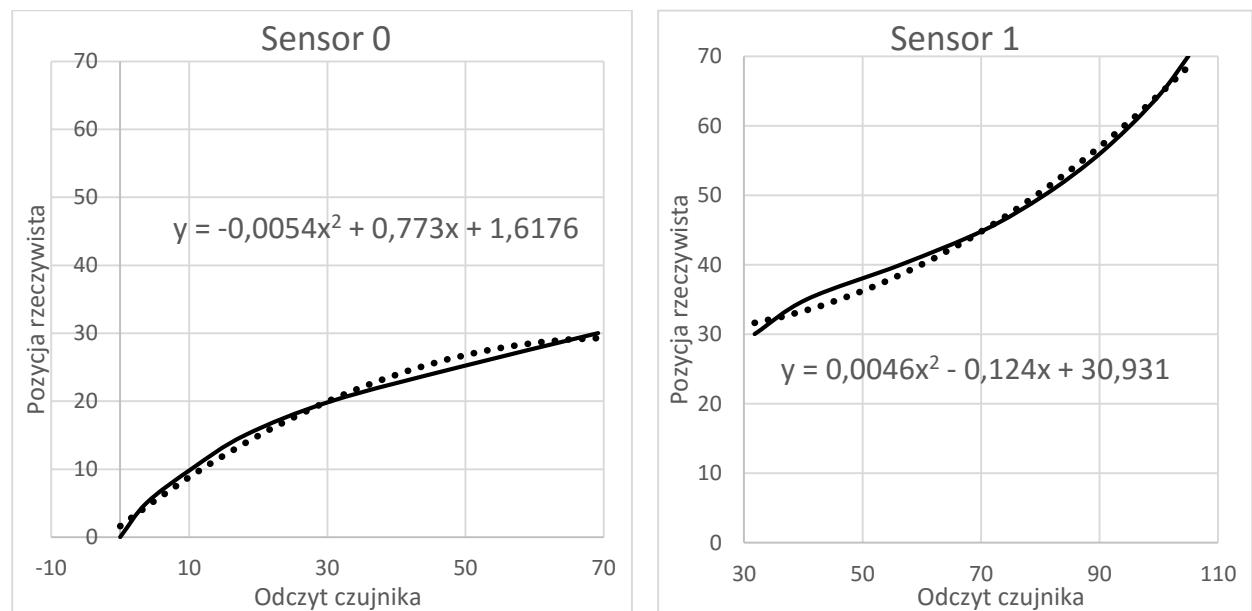
Rzeczywista pozycja kulki [mm]	Odczyt Sensor 0	Odczyt Sensor 1
0	<b>34</b>	957
50	<b>72</b>	949
100	<b>137</b>	939
150	<b>215</b>	911
200	<b>340</b>	890
250	<b>525</b>	833
300	<b>726</b>	<b>782</b>
350	832	<b>694</b>
400	900	<b>535</b>
450	903	<b>394</b>
500	942	<b>293</b>
550	956	<b>213</b>
600	964	<b>148</b>
650	970	<b>93</b>
700	922	<b>50</b>

Następnie zmodyfikowano wyniki pomiarów, tak, by pierwszy pomiar czujnika zgadzał się z rzeczywistą pozycją kulki. Od pomiarów Snsora 0 odjęto 34, następnie podzielono przez 10, uzyskując dystans w centymetrach. Natomiast wartości pomiarów Sensorsa 1 odjęto od 1100, wynik podzielono przez 10, również uzyskując dystans w centymetrach. Po zastosowaniu modyfikatorów otrzymano wyniki przedstawione w tabeli nr 4.

*Tabela 4. Zmodyfikowane pomiary wykonane czujnikami Pololu VL53L0X*

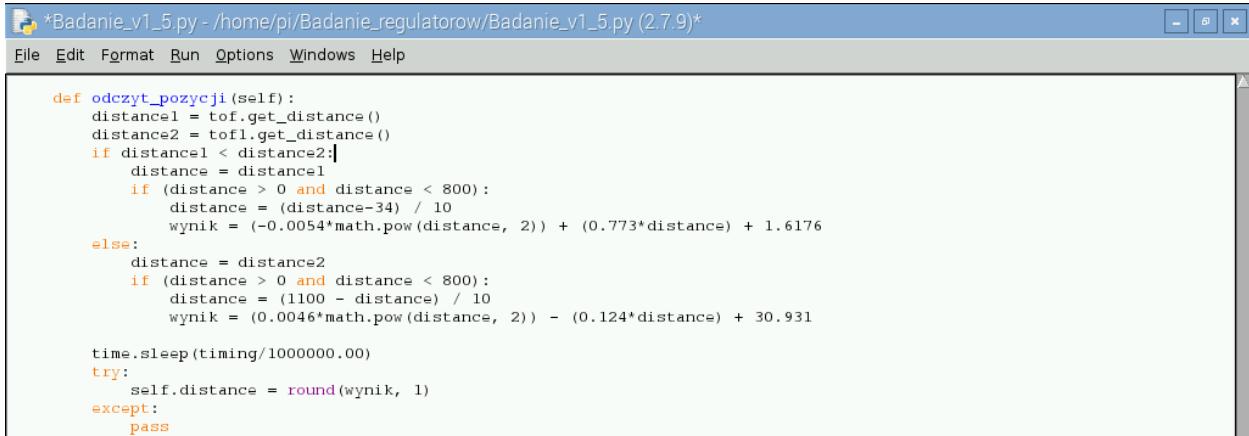
Rzeczywista pozycja kulki [cm]	(Sensor_0 – 34) / 10	(1100 – Sensor_1) / 10
0	<b>0</b>	
5	<b>3,8</b>	
10	<b>10,3</b>	
15	<b>18,1</b>	
20	<b>30,6</b>	
25	<b>49,1</b>	
30	<b>69,2</b>	<b>31,8</b>
35		<b>40,6</b>
40		<b>56,5</b>
45		<b>70,6</b>
50		<b>80,7</b>
55		<b>88,7</b>
60		<b>95,2</b>
65		<b>100,7</b>
70		<b>105</b>

W kolejnym kroku, w programie Microsoft Excel, utworzono wykresy zależności między zmodyfikowanymi odczytami z czujników do rzeczywistej pozycji kulki. Otrzymane wykresy przedstawiono na rysunku 50.



*Rys. 50. Wykresy zależności między zmodyfikowanymi odczytami czujników, a rzeczywistą pozycją kulki*

Do wykresów dodano linię trendu, która jest zbliżona do wykresu oraz wyświetlono wzór funkcji. Otrzymane wzory funkcji umożliwiają przeliczenie odczytów czujników na przybliżoną rzeczywistą pozycję kulki. Na rysunku 51 przedstawiono kod, który odczytuje pomiary z czujników, następnie oblicza przybliżoną rzeczywistą pozycję kulki na belce.



```

*Badanie_v1_5.py - /home/pi/Badanie_re regulatorow/Badanie_v1_5.py (2.7.9)*
File Edit Format Run Options Windows Help

def odczyt_pozycji(self):
    distance1 = tof.get_distance()
    distance2 = tof1.get_distance()
    if distance1 < distance2:
        distance = distance1
        if (distance > 0 and distance < 800):
            distance = (distance-34) / 10
            wynik = (-0.0054*math.pow(distance, 2)) + (0.773*distance) + 1.6176
    else:
        distance = distance2
        if (distance > 0 and distance < 800):
            distance = (1100 - distance) / 10
            wynik = (0.0046*math.pow(distance, 2)) - (0.124*distance) + 30.931
    time.sleep(timing/1000000.00)
    try:
        self.distance = round(wynik, 1)
    except:
        pass

```

Rys. 51. Kod służący do odczytu pomiarów z czujników

Ad. E. Następną funkcją są obliczenia regulatora PID. Wykorzystano do nich udostępnioną przez Python Software Fundation bibliotekę. W momencie uruchomienia układu, program odczytuje nastawy regulatora oraz wymuszenie wpisane przez użytkownika. Następnie uruchamiana jest funkcja odczytująca pozycję kulki. Zebrane dane przekazywane są do funkcji w bibliotece PID, w której dokonywane są obliczenia i zwracana wartość na wyjściu. Funkcja oblicza odpowiedź na podstawie omawianego wcześniej wzoru:

$$U(s) = K_P E(s) + K_I \frac{1}{s} E(s) + K_D s E(s) \quad (2.6)$$

Parametry  $K_P$ ,  $K_I$ ,  $K_D$ , to nastawy regulatora podane przez użytkownika, natomiast uchyb  $E(s)$  obliczany jest na podstawie aktualnych pomiarów pozycji kulki.

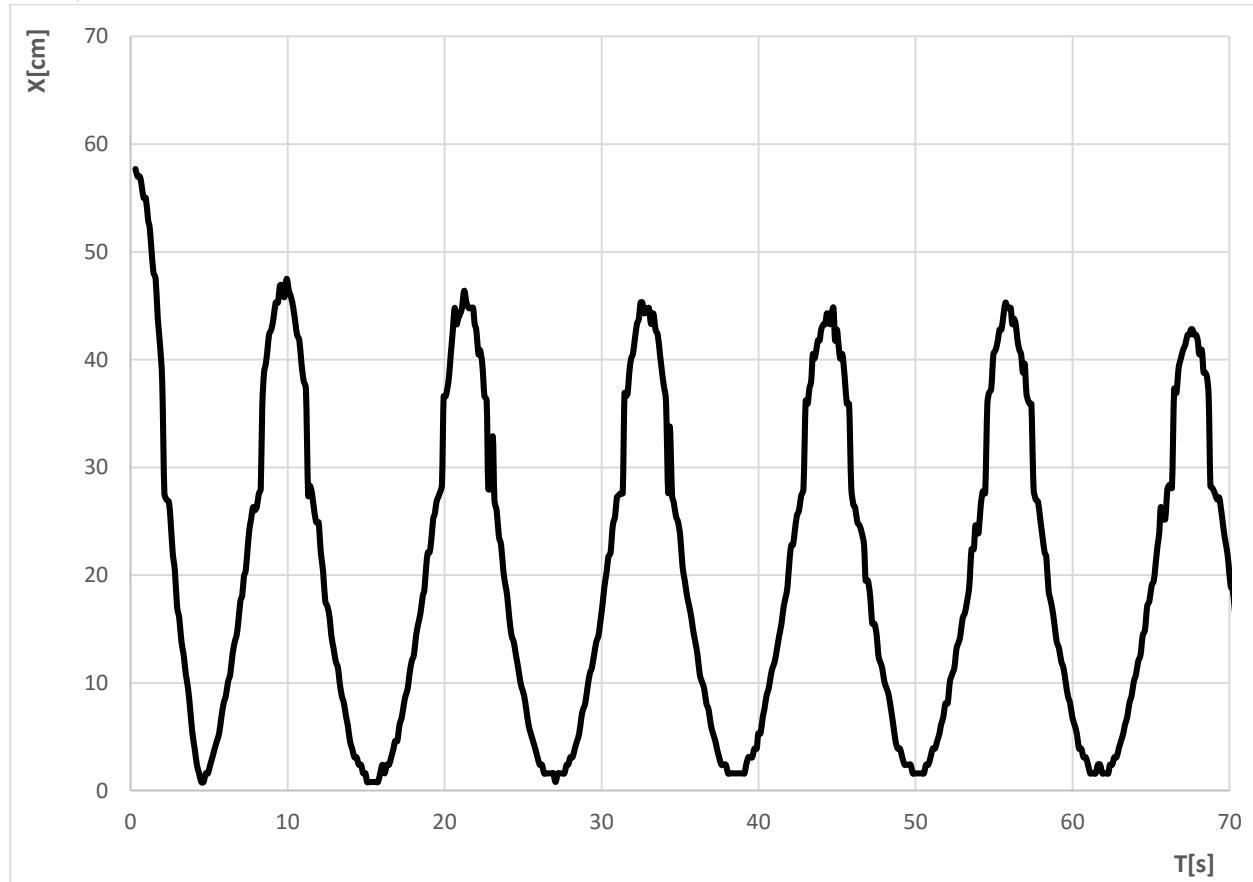
Ad. F. Wartość na wyjściu regulatora traktowana jest jako pożądaný kąt nachylenia belki w danej chwili. Układ porównuje pożądaný kąt nachylenia z kątem pod którym aktualnie nachylona jest belka. Wynikiem porównania jest decyzja o wykonaniu obrotu przez silnik. Określany jest kierunek oraz ilość kroków do wykonania. Obrót silnika jest wykonywany przez funkcję, w której zaprogramowano wysyłanie impulsów na 4 piny GPIO w odpowiedniej sekwencji. Po wykonaniu kroku modyfikowana jest wartość zmiennej, na podstawie której określany jest aktualny kąt nachylenia belki.

Wymienione powyżej funkcje zostały podzielone na 3 wątki. Wątki są to części programu wykonujące się jednocześnie. Do tworzenia i obsługi wątków służy specjalna biblioteka: „threading”. W celu umożliwienia przepływu informacji między wątkami konieczne jest zdefiniowanie kolejki do każdego wątku. Funkcja w ramach jakiegoś wątku odczytuje wartości znajdujące się w kolejce, które zostały wcześniej wrzucone do kolejki przez inne wątki. Użycie wątków było niezbędne do zapewnienia płynnej pracy urządzenia, ponieważ bez wątków, niektóre funkcje blokowały inne. Kolejność wykonywania funkcji wraz z podziałem na wątki, oraz algorytm obsługujący wątki przedstawia schemat stanowiący załącznik nr 4. Natomiast pliki programu sterującego pracą całego układu dodano w załączniku nr 5.

### 3. BADANIE REGULATORA

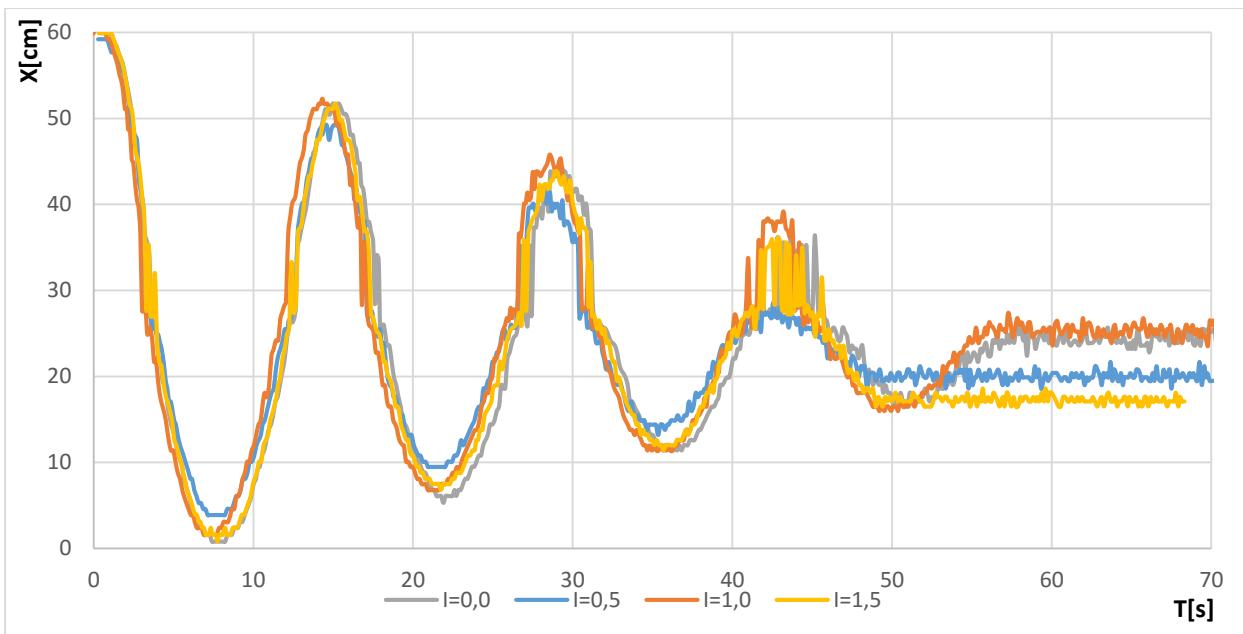
#### 3.1. Strojenie ręczne

Początkowa pozycja kulki ustalona została na pozycji  $X_0=60\text{cm}$ . Jako wymuszenie skokowe ustawiano wartość  $X_U=20\text{cm}$ . Nastawy  $K_I$  oraz  $K_D$  ustawiono na zero. Natomiast wzmacnienie  $K_p$  ustawiono na wartość 1. Uruchomiono układ obserwując wykres zmiany położenia kulki w czasie. Parametr  $K_p$  zwiększano co 1. Po zadziałaniu wymuszenia oscylacje kulki spadały. Dopiero przy wzmacnieniu  $K_p = 8$  oscylacje lekko rosły. Następnie zmniejszano wartość  $K_p$  o 0.1, aby uzyskać wzmacnienie graniczne. Zauważono, iż przy wartości  $K_p = 7,5$  uzyskano oscylację o podobnej amplitudzie w każdym cyklu. Natomiast przy wartości  $K_p = 7,4$  amplituda lekko spadała. W związku z tym uznano iż wartość graniczna wzmacnienia  $K_u = 7,5$ . Na rysunku 52 przedstawiono wykres pozycji kulki od czasu przy nastawach regulatora  $K_p=7,5$ ;  $K_I=0$ ;  $K_D=0$ .



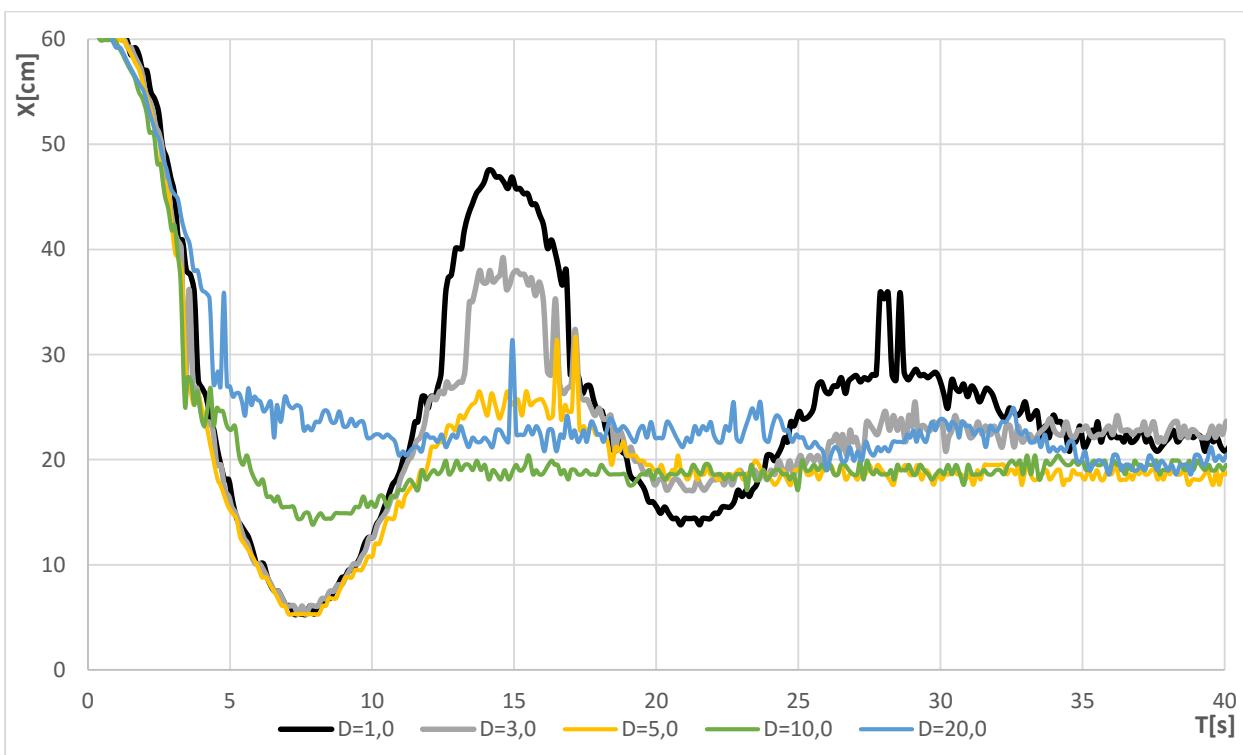
Rys. 52. Wykres pozycji kulki od czasu przy nastawach regulatora:  $K_p=7,5$ ;  $K_I=0$ ;  $K_D=0$

Po ustaleniu wzmacnienia granicznego, przyjęto nastawę wzmacnienia części proporcjonalnej jako połowę wzmacnienia granicznego. Następnie zwiększano nastawę wzmacnienia części całkującej zaczynając od zera, zwiększając o wartość 0,5. Tym sposobem otrzymano wykresy zaprezentowane na rysunku nr 53.



Rys. 53. Wykres pozycji kulki od czasu przy różnych nastawach części całkującej  $K_I$  ( $I$ ) i stałych nastawach regulatora:  $K_p=3,75$ ;  $K_D=0$

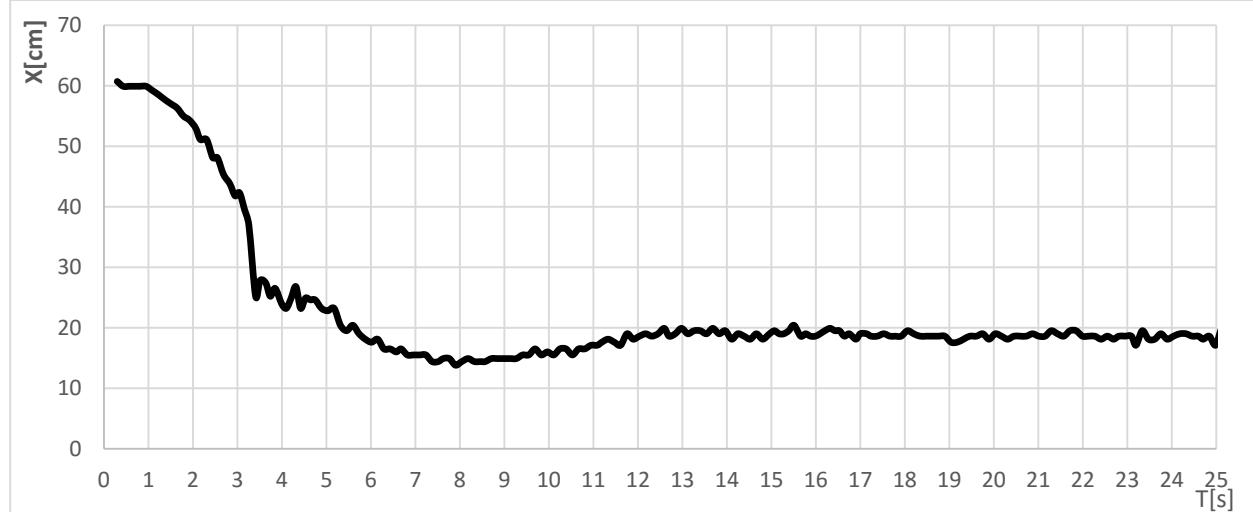
Z analizy wykresów wynika, iż najlepszą dokładność w stanie ustalonym układ osiąga przy wartości  $K_I=0,5$ . Ostatnim etapem strojenia ręcznego jest znalezienie odpowiedniej wartości wzmocnienia różniczkującego  $K_D$ . W tym celu zwiększano wartość  $K_D$ . Na rysunku 54. przedstawiono wykresy dla różnych nastaw wzmocnienia  $K_D$ , przy stałych nastawach  $K_p=3,75$ ;  $K_I=0,5$ .



Rys. 54. Wykres pozycji kulki od czasu przy różnych nastawach części różniczkującej  $K_D$  ( $D$ ) i stałych nastawach regulatora:  $K_p=3,75$ ;  $K_I=0,5$

Po analizie wyników pomiarów pracy regulatora przy różnych nastawach wzmacnienia różniczkowego, stwierdzono, że układ pracuje najlepiej przy wzmacnieniu o wartości 10. Przy wyższym wzmacnieniu różniczkowym, układ jest mniej stabilny.

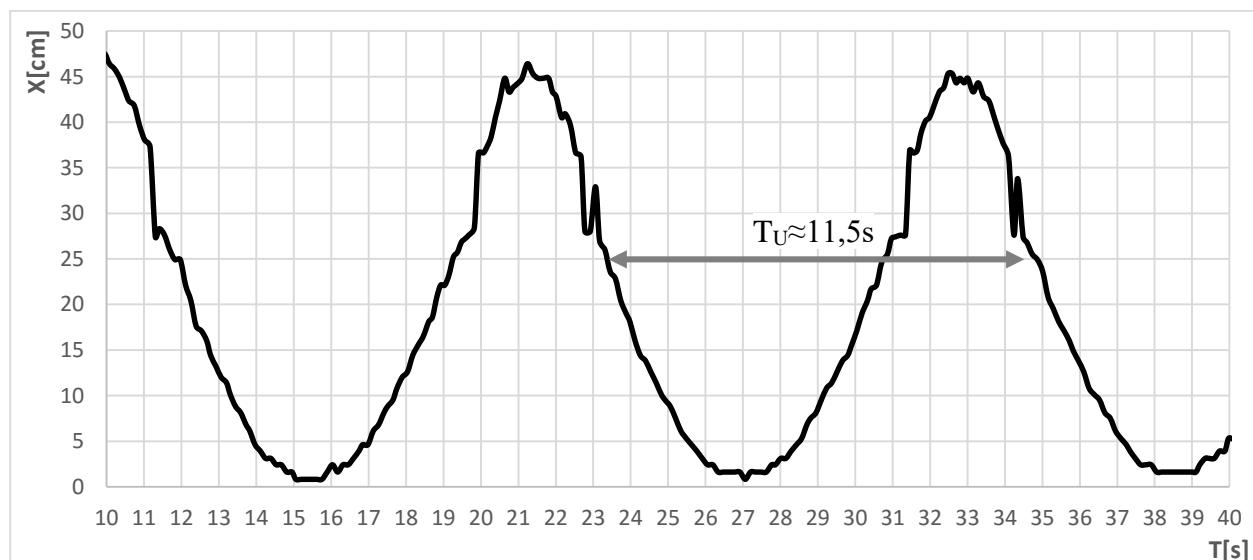
Metodą strojenia ręcznego uzyskano nastawy:  $K_P=3,75$ ;  $K_I=0,5$ ;  $K_D=10$ . Przy takich parametrach, pozycji początkowej kulki  $X_0=60\text{cm}$  i wymuszeniu skokowym  $X_U=20\text{cm}$ , układ osiąga cel po ok. 12 sekundach, co można zaobserwować na rysunku 55.



Rys. 55. Wykres pozycji kulki od czasu przy różnych nastawach regulatora:  $K_P=3,75$ ;  $K_I=0,5$ ;  $K_D=10$

### 3.2 Wyznaczanie nastaw metodą Zieglera-Nicholsa

Wyznaczanie nastaw metodą Zieglera-Nicholsa również wymaga znalezienie wzmacnienia granicznego przy którym obserwowane są niegasnące oscylacje. Ponadto należy wyznaczyć okres oscylacji. Strojenie metodą Zieglera-Nicholsa przeprowadzane zachowując takie same warunki. Pozycja wyjściowa kulki  $X_0=60\text{cm}$ , wymuszenie skokowe  $X_U=20\text{cm}$ . Wartość wzmacnienia granicznego została wyznaczona podczas strojenia ręcznego i wynosi  $K_U=7,5$ . Natomiast okres oscylacji wyznaczono na podstawie wykresu przedstawionego na rysunku nr 56.



Rys. 56. Fragment wykresu pozycji kulki od czasu przy nastawach regulatora:  $K_P=7,5$ ;  $K_I=0$ ;  $K_D=0$

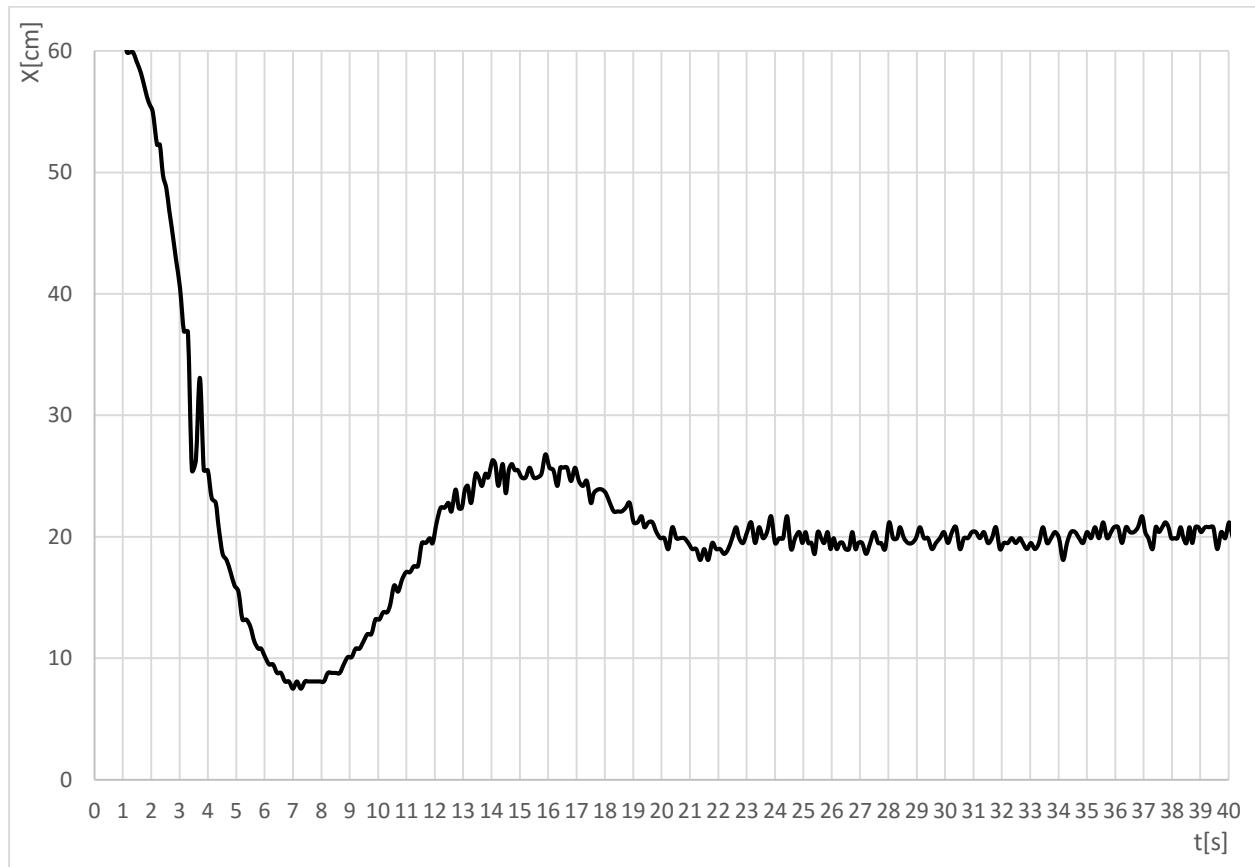
Na podstawie wykresu wyznaczono okres oscylacji  $T_U \approx 11,5$ s. Mając te dane obliczono nastawy regulatora równaniami podanymi w metodzie Zieglera-Nicholsa:

$$K_P = 0,6K_U = 4,5 \quad (3.1)$$

$$K_I = \frac{1,2K_U}{T_U} \approx 0,78 \quad (3.2)$$

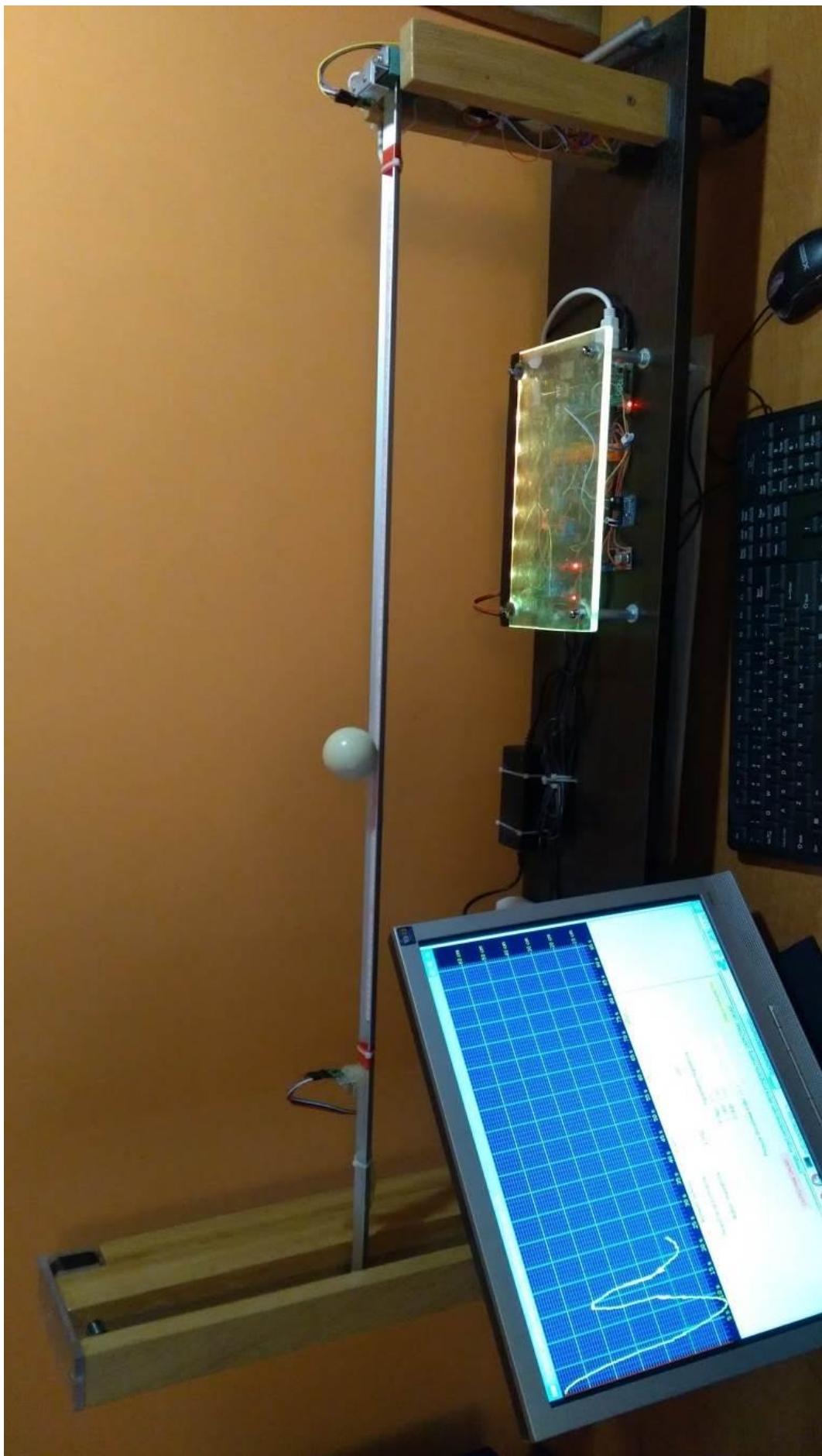
$$K_D = \frac{0,6K_UT_U}{8} \approx 6,47 \quad (3.3)$$

Następnie przeprowadzono badanie wykorzystując obliczone powyżej wyniki nastaw. Wykres przedstawiający pozycje kulki od czasu, przy parametrach regulatora  $K_P=4,5$ ;  $K_I=0,78$ ;  $K_D=6,47$  przedstawia rysunek 57.



Rys. 57. Fragment wykresu pozycji kulki od czasu przy nastawach

W trakcie pracy z nastawami wyznaczonymi metodą Zieglera-Nicholsa układ osiągnął cel po ok. 20 sekundach. Praca układu jest stabilna. Na rysunku 58 pokazano urządzenie w trakcie pracy.



Rys. 58. Układ kulka-belka w trakcie pracy

Podsumowując strojenie ręczne, jak i wyznaczanie nastaw metodą Ziegera-Nicholsa umożliwiły znalezienie wzmacnienia poszczególnych członów regulatora PID, zapewniających stabilną pracę układu kulka-belka. Parametry wyznaczone podczas strojenia ręcznego pozwoliły na efektywniejszą pracę układu niż nastawy wyznaczone metodą Ziegera-Nicholsa. Jednakże proces strojenia metodą Ziegera-Nicholsa był dużo szybszy, ponieważ nie wymaga wykonania tak wielu doświadczeń. Na wyniki badań miały wpływ niedokładne i niestabilne pomiary pozycji, niewielkie wyginanie belki pod ciężarem kuli, charakter pracy silnika krokowego powodujący drgania układu oraz program, którego algorytmy można zoptymalizować.

## ZAKOŃCZENIE

Celem niniejszej pracy było zbudowanie urządzenia pozwalającego na badanie regulatora PID oraz przeprowadzenie badań, których wynikiem są nastawy wzmacnienia członu proporcjonalnego, całkującego oraz różniczkującego. Cel pracy został osiągnięty. Wszystkie cele szczegółowe zostały zrealizowane. Wykonano projekt układu kulka-belka w programie SolidWorks, następnie zrealizowano projekt rozwiązując problemy konstrukcyjne oraz niedoskonałość urządzeń wchodzących w skład układu. W trakcie budowy układu kulka-belka sprawdzono różne metody i czujniki służące do wyznaczenia aktualnej pozycji kulki na belce oraz nachylenia belki. Po testach zdecydowano, że najlepszym rozwiązaniem do pomiaru pozycji kulki jest użycie dwóch cyfrowych czujników optycznych Pololu VL53L0X; natomiast do wyznaczania nachylenia belki, żyroskop ze względu na wadliwą pracę zastąpiono funkcją w programie zliczającą kroki silnika krokowego. Następnie przedstawiono moduły wchodzące w skład układu. W pełni zaprogramowano układ, tworząc intuicyjny interfejs graficzny z możliwością wyświetlania wykresu w czasie rzeczywistym. Stworzono algorytm oparty na trzech niezależnie pracujących wątkach, wymieniających ze sobą informacje. Na zbudowanym urządzeniu przeprowadzono szereg badań. Mimo niedokładnych metod pomiarowych i zmiennych warunków otoczenia, udało się uzyskać prawidłową i stabilną pracę układu. Do wyznaczenia nastaw regulatora PID wykorzystano dwie metody doświadczalne, uzyskując nastawy zapewniające efektywną i stabilną pracę układu. Metodą Ziegera-Nicholsa otrzymano nastawy:  $K_P=4,5$ ;  $K_I=0,78$ ;  $K_D=6,47$ , natomiast wynikiem strojenia ręcznego są nastawy:  $K_P=3,75$ ;  $K_I=0,5$ ;  $K_D=10$ . Nastawy uzyskane poprzez strojenie ręczne zapewniają efektywniejszą pracę układu, jednak metoda ta jest bardziej pracochłonna, wymaga przeprowadzenie dużo większej liczby doświadczeń.

Kluczowym dla realizacji pracy było rozwiązanie kilku ważnych problemów:

- niestabilne pomiary pozycji kulki na belce - zastosowanie czujnika odległości, o zakresie pracy 2-200cm nie dawało stabilnych pomiarów na całej długości belki. Problem rozwiązano dodając drugi czujnik po przeciwej stronie i skracając odcinek na którym kulka może się poruszać ze 100cm na 70cm;
- nieprawidłowe pomiary pozycji kulki na belce - odczyty uzyskane z zastosowanych czujników odległości nie pokrywały się z rzeczywistością. Skorygowano problem korelując odczyty ze zmierzonymi, rzeczywistymi pozycjami kulki na belce. Następnie znaleziono funkcje matematyczne opisujące zależności między tymi danymi. Na podstawie uzyskanych funkcji zaimplementowano funkcję, która przelicza odczyt z czujnika na przybliżoną, rzeczywistą pozycję kulki;
- implementacja wykresu wyświetlającego wyniki w czasie rzeczywistym – w języku programowania Python brakuje bibliotek lub modułów zawierających gotowe narzędzie do wyświetlania wykresów w czasie rzeczywistym. Wyświetlanie wykresu rozwiązano wykorzystując narzędzie do rysowania prostych figur geometrycznych. Wykres rysowany jest poprzez dodawanie krótkich odcinków;
- nieprawidłowy pomiar nachylenia belki - na zaimplementowanym wykresie wyświetlającym wyniki w czasie rzeczywistym zauważono nieprawidłowe odczyty z żyroskopu, co mocno zakłócało pracę całego układu. Odczyty żyroskopu zastąpiono funkcją w programie, która zlicza kroki silnika, a następnie przelicza na kąt pochylenia belki;
- zbyt długi czas wykonywania funkcji w programie – funkcja odpowiedzialna za wysyłanie odpowiednich sygnałów do silnika krokowego potrzebuje ok. 20ms do wykonania pełnej sekwencji kroków, pozostałe funkcje programu również

wykonywane są w łącznym czasie ok 20ms. Wykonanie całej pętli ze wszystkimi niezbędnymi instrukcjami było zbyt wolne, aby efektywnie sterować dynamicznym obiektem. Problem rozwiązano stosując wielowątkowość. Cały program podzielono na trzy wątki wykonywające jednocześnie różne funkcje. Zapewniło to dużo szybszą reakcję na zmianę pozycji kulki na belce i płynniejszą pracę układu.

Zbudowany układ kulka-belka można rozbudować o wiele różnych modułów rozszerzających możliwości urządzenia. Na układzie można badać również inne regulatory dodając do programu nowe funkcje.

Efektywność pracy układu można podnieść między innymi poprzez:

- zastosowanie innych czujników, lub innej metody pomiaru pozycji kulki na pochylni dającej bardziej dokładne i stabilne odczyty,
- użycie innej metody lub czujnika do wyznaczania nachylenia belki,
- optymalizację programu, skracając algorytmy można zwiększyć szybkość odpowiedzi układu na zmiany warunków,
- zastosowanie szybszego serwomechanizmu zamiast silnika krokowego,
- wyznaczenie innych nastaw regulatora np. inną metodą.

Niniejsza praca stanowi tylko aplikacje jednego z zastosowań regulatora PID w układzie sterowania i przedstawiony materiał nie wyczerpuje oczywiście całego tematu związanego z zagadnieniem. Co może być podstawą do opracowania innych ciekawych metod sterowania, w tym metod sterowania optymalnego lub suboptymalnego.

## BIBLIOGRAFIA

1. J. Awrejcewicz, W. Wodzicki, Podstawy automatyki. Teoria i przykłady, Wydawnictwo Politechniki Łódzkiej, Łódź 2001
2. J. Wicher, Bezpieczeństwo samochodów i ruchu drogowego, Wydawnictwo Komunikacji i Łączności WKŁ, Warszawa 2012
3. A. Dębowksi, Automatyka. Podstawy teorii, Wydawnictwo Naukowe PWN, Łódź 2007
4. G. L. de Oliveira Serra, Frontiers in Advanced Control Systems, InTech, Rijeka 2012
5. <https://encyklopedia.pwn.pl/haslo/sprzezenie-zwrotne;3978598.html>
6. K. Astrom, T. Hagglund, PID Controllers: Theory, Design, and Tuning, Instrument Society of America, Research Triangle Park 1995
7. J. Brzózka, Regulatory i układy automatyki, Wydawnictwo MIKOM, Warszawa 2004
8. Z. Bubnicki, Teoria i algorytmy sterowania, Wydawnictwo Naukowe PWN, Warszawa 2005
9. Z. Skup, Podstawy automatyki i sterowania, Wydawnictwo Politechniki Warszawskiej, Warszawa 2012
10. Li, Y. and Ang, K.H. and Chong, G.C.Y. (2006) PID control system analysis and design. IEEE Control Systems Magazine 26(1):pp. 32-41.
11. <http://www.solidworks.pl/> (dostęp 03.02.2017)
12. <http://www.sklep-lager.pl/lozyska-kulkowe/115-lozysko-kulkowe-jednorzedowe.html> (dostęp 03.02.2017)
13. [http://sklep.alukwas.pl/1983-home\\_default/katownik-aluminiowy-50x50x3dlug-05mb.jpg](http://sklep.alukwas.pl/1983-home_default/katownik-aluminiowy-50x50x3dlug-05mb.jpg) (dostęp 07.06.2017)
14. <http://rab.ict.pwr.wroc.pl/~kreczmer/wds/prezentacje/dalmierze-optyczne.pdf> (dostęp 08.06.2017)
15. <http://www.wobit.com.pl/produkt/1076/silniki-krokowe-dwufazowe/42bygh118-01b/> (dostęp 08.06.2017)
16. [http://img.dxcdn.com/productimages/sku\\_408436\\_1.jpg](http://img.dxcdn.com/productimages/sku_408436_1.jpg) (dostęp 10.09.2017)
17. <https://botland.com.pl/sterowniki-silnikow-moduly/3164-l298-dwukanalowy-sterownik-silnikow-modul.html> (dostęp 10.09.2017)
18. <http://www.printo3d.pl/wp-content/uploads/2015/08/4.jpg> (dostęp 10.09.2017)
19. [http://media.rs-online.com/t\\_large/F8111284-01.jpg](http://media.rs-online.com/t_large/F8111284-01.jpg) (dostęp 10.09.2017)
20. <https://botland.com.pl/moduly-i-zestawy-raspberry-pi-a-i-b/2543-raspberry-pi-model-bplus-512mb-ram.html> (dostęp 12.09.2017)
21. <https://www.raspbian.org/> (dostęp 15.09.2017)
22. <https://cdn3.volusion.com/btfzd.umflq/v/vspfiles/photos/AD281-2.jpg?1475058830> (dostęp 08.11.2017)
23. <https://botland.com.pl/przetwornice-step-down/2967-przetwornica-step-down-lm2596-15v-35v-3a.html> (dostęp 08.11.2017)
24. <https://botland.com.pl/przekazniki/2170-dwukanalowy-modul-przekaznikow-rm5-z-izolacja-optoelektroniczna-5v-10a125vac.html> (dostęp 08.11.2017)
25. [http://technovade.pl/media/catalog/product/cache/1/image/650x/040ec09b1e35df139433887a97daa66f/r/m/rm-2\\_0.jpg](http://technovade.pl/media/catalog/product/cache/1/image/650x/040ec09b1e35df139433887a97daa66f/r/m/rm-2_0.jpg) (dostęp 08.11.2017)
26. <http://www.python.rk.edu.pl/w/p/wprowadzenie-do-pythona/> (dostęp 15.09.2017)

## SPIS TABEL

Tabela 1. Wybrane parametry silnika krokowego, bipolarnego 42BYGH118-01 .....	29
Tabela 2. Parametry modulu sterownika L298N.....	30
Tabela 3. Pomiary wykonane czujnikami Pololu VL53L0X .....	38
Tabela 4. Zmodyfikowane pomiary wykonane czujnikami Pololu VL53L0X.....	39

## SPIS RYSUNKÓW

Rys. 1. Ogólna struktura regulatora PID .....	11
Rys. 2. Szkic części tworzonej w programie SolidWorks Premium 2013 .....	12
Rys. 3. Model wałka 3D .....	12
Rys. 4. Model wałka 3D z zaokrąglonymi krawędziami .....	13
Rys. 5. Wycięcie otworu w bryle .....	13
Rys. 6. Gotowy rysunek techniczny wygenerowany w programie.....	14
Rys. 7. Wiązanie wałka z łożyskami, oraz otworem pod łożysko .....	14
Rys. 8. Pełen model mechaniczny układu kulka-belka .....	15
Rys. 9. Projekt uchwytu belki przy niższym ramieniu .....	15
Rys. 10. Projekt specjalnego pręta kwadratowego .....	16
Rys. 11. Złożenie uchwytu z prętem osadzonym na łożyskach w otworach .....	16
Rys. 12. Projekt uchwytu belki przy wyższym ramieniu .....	17
Rys. 13. Układ wciągający linkę z zamocowaną belką 1 – walek, 2 – łożyska, 3 – sprzęgło kłowe (w uproszczeniu), 4 – silnik krokowy (w uproszczeniu) .....	17
Rys. 14. Drewniana konstrukcja urządzenia .....	18
Rys. 15. Łożysko kulkowe 605 ZZ umieszczone w wyfrezowanym otworze .....	18
Rys. 16. Sprzęgło kłowe z wkładką elastyczną .....	19
Rys. 17. Silnik krokowy bipolarne 42BYGH118-01 ze sprzęgłem kłowym .....	19
Rys. 18. Mocowanie silnika oraz wałka z nawiniętą linką.....	20
Rys. 19. Uchwyt belki od strony silnika .....	20
Rys. 20. Uchwyt zamontowany między ramionami .....	21
Rys. 21. Kątownik aluminiowy .....	21
Rys. 22. Biała bila na belce.....	22
Rys. 23. Uchwyt belki .....	22
Rys. 24. Schemat budowy urządzenia .....	23
Rys. 25. Prototyp układu kulka belka do badania regulatora .....	24
Rys. 26. Schemat pomiaru pozycji kulki.....	24
Rys. 27. Projekt belki z taśmą oporową i miedzianą .....	25
Rys. 28. Czujnik ultradźwiękowy HC-SR04 .....	25
Rys. 29. Analogowy czujnik optyczny Sharp GP2Y0A21YK0F .....	26
Rys. 30. Schemat pomiaru odległości dalmierzem na zasadzie triangulacji .....	26
Rys. 31. Wykres pozycji kulki w czasie pracy układu - pomiar dalmierzami Sharp GP2Y0A21YK0F .....	27
Rys. 32. Czujnik optyczny Pololu VL53L0X.....	28
Rys. 33. Wykres pozycji kulki w czasie pracy układu - pomiar dalmierzami Pololu VL53L0X....	28
Rys. 34. Moduł MPU-6050 .....	29
Rys. 35. Silnik bipolarne 42BYGH118-01 .....	29
Rys. 36. Obliczenie dokładności sterowania nachyleniem belki.....	30
Rys. 37. L298N - dwukanałowy sterownik silników.....	30

<i>Rys. 38. Schemat silnika krokowego bipolarnego, 4-ro przewodowego</i> .....	31
<i>Rys. 39. Minikomputer Raspberry Pi B+</i> .....	31
<i>Rys. 40. Połączone moduły urządzenia</i> .....	32
<i>Rys. 41. Płyta ochronna z sygnalizacją świetlną</i> .....	33
<i>Rys. 42. Moduł przetwornicy step-down LM2596</i> .....	33
<i>Rys. 43. Dwukanałowy moduł przekaźników RM5</i> .....	34
<i>Rys. 44. Kod napisany w edytorze skryptów – Python IDLE</i> .....	34
<i>Rys. 45. Komenda uruchamiająca program, wpisana w terminalu Raspberry Pi</i> .....	35
<i>Rys. 46. Fragment kodu, budującego interfejs graficzny użytkownika</i> .....	36
<i>Rys. 47. Interfejs graficzny zaprogramowany w języku Python przy użyciu biblioteki Tkinter</i> .....	36
<i>Rys. 48. Fragment kodu wykonującego rysowanie wykresu</i> .....	37
<i>Rys. 49. Fragment pliku z zarejestrowanym pomiarem</i> .....	38
<i>Rys. 50. Wykresy zależności między zmodyfikowanymi odczytami czujników, a rzeczywistą pozycją kulki</i> .....	39
<i>Rys. 51. Kod służący do odczytu pomiarów z czujników</i> .....	40
<i>Rys. 52. Wykres pozycji kulki od czasu przy nastawach regulatora: <math>K_P=7,5</math>; <math>K_I=0</math>; <math>K_D=0</math></i> .....	41
<i>Rys. 53. Wykres pozycji kulki od czasu przy różnych nastawach części całkująccej <math>K_I</math> (I) i stałych nastawach regulatora: <math>K_P=3,75</math>; <math>K_D=0</math></i> .....	42
<i>Rys. 54. Wykres pozycji kulki od czasu przy różnych nastawach części różniczkującej <math>K_D</math> (D) i stałych nastawach regulatora: <math>K_P=3,75</math>; <math>K_I=0,5</math></i> .....	42
<i>Rys. 55. Wykres pozycji kulki od czasu przy różnych nastawach regulatora: <math>K_P=3,75</math>; <math>K_I=0,5</math>; <math>K_D=10</math></i> .....	43
<i>Rys. 56. Fragment wykresu pozycji kulki od czasu przy nastawach regulatora: <math>K_P=7,5</math>; <math>K_I=0</math>; <math>K_D=0</math></i> .....	43
<i>Rys. 57. Fragment wykresu pozycji kulki od czasu przy nastawach</i> .....	44
<i>Rys. 58. Układ kulka-belka w trakcie pracy</i> .....	45

## SPIS ZAŁĄCZNIKÓW

1. Załącznik Rysunki wykonawcze części wygenerowane w programie SolidWorks (na płycie DVD)
2. Załącznik Pliki trójwymiarowego modelu układu kulka-belka skonstruowanego w programie SolidWorks (na płycie DVD)
3. Załącznik Schemat elektryczny układu kulka-belka
4. Załącznik Schemat algorytmów programu
5. Załącznik Pliki programu sterującego pracą układu kulka belka napisany w języku programowania Python (na płycie DVD)